

Lenguaje Específico de Dominio para Aplicaciones de Modelación Glaciológica

Matías Gel¹, Adriana Urciuolo¹, Rodolfo Iturraspe¹,

¹ Universidad Nacional de Tierra del Fuego, IDEI.
Onas 450, (9410) Ushuaia, Argentina
{[mgel](mailto:mgel@untdf.edu.ar), [aurciuolo](mailto:aurciuolo@untdf.edu.ar), [riturraspe](mailto:riturraspe@untdf.edu.ar)}@untdf.edu.ar

Abstract. Las aplicaciones de software de modelado de la naturaleza se caracterizan por la complejidad propia de la información y procesos ambientales. Se requiere un nivel de abstracción adecuado y flexibilidad para la modelación de los procesos naturales. Algunas limitaciones del software de dominio provienen del uso de herramientas y lenguajes que carecen de facilidades para representar los conceptos particulares de estas aplicaciones. El Modelado específico del dominio (DSM) propone elevar el nivel de abstracción, expresando las soluciones directamente en términos del dominio. En este trabajo se presenta un lenguaje específico de dominio para aplicaciones de modelación del comportamiento de glaciares, y un generador de código para la transformación de los modelos en aplicaciones utilizables por los expertos.

Keywords: lenguaje, específico, dominio, mde, glaciología.

1 Introducción

Los sistemas que modelan el comportamiento de la naturaleza (agua, clima, etc.) constituyen dominios con conceptos y funcionalidades específicas que se caracterizan por la complejidad propia del manejo de información ambiental [1]. Se incluyen en este dominio aplicaciones de software orientadas a la simulación de procesos físicos, optimización y soporte de decisión [2], tales como los modelos de simulación climáticos, hidrológicos [3], etc. Si bien se han realizado avances tendientes al manejo de la complejidad de estas aplicaciones (tales como el diseño de frameworks [4]), a menudo presentan problemas para los usuarios expertos del dominio, vinculados con la falta de flexibilidad para modelar los sistemas naturales en distintas configuraciones de componentes, así como para la integración de las distintas etapas del desarrollo de estos sistemas: la captura y pre - procesamiento de observaciones, el modelado y simulación de los procesos físicos y la presentación de la información producida en formatos adecuados para expertos [5]. Algunas de las limitaciones de este software provienen del uso de lenguajes de propósito general, los cuales resultan complejos de utilizar por expertos en un dominio de estas características. Tampoco cuentan con facilidades para trabajar en un nivel de abstracción apropiado a las particularidades de sus conceptos y problemas. Se afirma que el modelado ambiental está atrapado por herramientas que no se adaptan a sus requerimientos [2].

El Modelado específico de Dominio (DSM) permite elevar la capacidad de abstracción más allá del código, especificando programas que utilizan directamente los conceptos del dominio. Los productos finales pueden luego generarse a partir de dichas especificaciones de alto nivel [6]. Mediante el uso de lenguajes específicos del dominio (DSL), se logra que los desarrolladores y usuarios utilicen una sintaxis común al área en que trabajen. Los DSL creados en cooperación con expertos del dominio pueden alcanzar calidad superior y responder en forma ajustada a su propósito [7]. En una solución DSM, los DSL se complementan con generadores de código que transforman los modelos en código final [8], los cuales normalmente se ejecutan con el soporte de un framework de aplicación específico del dominio

Para poder alcanzar su objetivo, una solución DSM debe focalizarse en un dominio acotado. Por ello y considerando la variabilidad del dominio de modelación y simulación ambiental, el presente trabajo se enfoca en una familia particular del software de modelación hidrológica: las aplicaciones de modelación glaciológica. Se presenta un entorno integrado de desarrollo basado en DSM creado con un enfoque de ingeniería dirigida por modelos, para dar soporte a estas aplicaciones, facilitando el trabajo del desarrollador de software en la creación de aplicaciones para el monitoreo y cálculo de los diferentes procesos físicos, así como en las tareas de gestión y presentación de la información. Se analiza la arquitectura de la solución DSM utilizada y sus distintos componentes: DSL, generador de código, framework específico de dominio. No obstante la importancia de este último en el marco de la solución DSM, el presente trabajo se focaliza en el desarrollo del DSL, incluyendo una aplicación que ejemplifica su utilización.

2 Arquitectura de Solución DSM para Modelación Glaciológica

Existen distintos enfoques para el desarrollo de DSLs [9]. En el presente trabajo se ha optado por una forma típica de creación de entornos DSM, donde se parte de desarrollar modelos sobre la base de conceptos aplicados por expertos del dominio, utilizando los mismos como input para generadores de código según los principios fundamentales de MDE (Ingeniería dirigida por modelos) [10].

Para el desarrollo de una solución DSM en un contexto MDE, se utiliza una arquitectura de 3 niveles sobre la plataforma de ejecución: un lenguaje específico de dominio, un generador de código y un framework específico de dominio. Siguiendo este enfoque, la Fig. 1 muestra la arquitectura propuesta para el DSM.

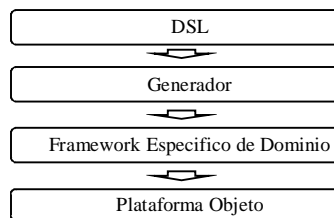


Fig. 1. Arquitectura utilizada para el entorno DSM

2.1 Elementos del DSM

En el marco de DSM, el DSL es la herramienta que permite abstraer la complejidad en un dominio dado, a través de la provisión de conceptos y reglas expresados directamente en el dominio del problema. Un DSL se define como “un lenguaje de programación de limitada expresividad focalizado en un dominio en particular” [11].

Los conceptos principales suelen tener su representación en la notación del lenguaje (gráfica o textual), otros conceptos se representan mediante conexiones o propiedades. En este trabajo se propone una notación gráfica; el lenguaje se formalizó con un meta-modelo que define su sintaxis abstracta. Sobre él se definió una sintaxis concreta con la cual el experto del dominio interactúa.

El generador de código especifica cómo se extrae la información de los modelos y se transforma en código ejecutable que no necesita ser modificado para su funcionamiento.

El Framework específico de dominio es una interface entre el código generado y la plataforma subyacente. Su objetivo es simplificar la tarea del generador, elimina la duplicación de código y provee de una capa de abstracción sobre la plataforma. Al proveer de componentes previamente testeados, se permite un aseguramiento de la calidad del producto. La plataforma comprende el conjunto de sistema operativo, máquina virtual, lenguajes, librerías y frameworks de componentes.

3 Implementación de un DSM para aplicaciones de glaciología

3.1. Análisis del Dominio

El análisis de dominio se realizó en forma conjunta con los expertos, a los fines de obtener el conocimiento necesario sobre sus conceptos y procesos centrales. Como resultado se obtuvo un Modelo de Dominio (base para la sintaxis abstracta), donde se representan las principales abstracciones utilizadas en las distintas etapas de estos sistemas, mediante un Diagrama de clases de Dominio UML. En cuanto a los procesos físicos a modelar, el presente trabajo se enfoca en el denominado “balance de masa”: diferencia entre la cantidad de agua que el glaciar pierde (por derretimiento) o gana (por acumulación) durante un periodo de tiempo. Para el cálculo del mismo se realizan observaciones periódicas en la superficie del glaciar.

3.2. Desarrollo del DSL

Sintaxis abstracta. El Metamodelo que se muestra en la Fig. 2 constituye la *sintaxis abstracta* del DSL, la cual se implementó mediante facilidades provistas por EMF (Eclipse Modeling Framework), en particular las herramientas Ecore, Emf.edit y Emf.genmodel, utilizadas para la descripción de la sintaxis y creación de editores, así como para la persistencia del meta-modelo [12].

Cada una de las clases del dominio tiene su derivación en una Eclass de ecore.

El meta-modelo contiene los componentes necesarios para la representación gráfica del DSL y para la generación de código. Captura las abstracciones necesarias

La herramienta seleccionada para la descripción de la sintaxis concreta fue un plugin para Eclipse denominado Obeo Designer. La especificación del DSL se realizó en un entorno que combina herramientas gráficas con expresiones en el lenguaje Aceleo 3, una extensión del lenguaje OCL de OMG.

Construcción del IDE. Para la construcción de un lenguaje visual, es necesario proveer un entorno de diseño que ofrezca paleta de componentes visuales, área de trazado, etc. La construcción del lenguaje gráfico se realizó en un proceso iterativo partiendo de la definición de las construcciones principales, refinando en pasos siguientes con el resto de los elementos. Los pasos realizados fueron los siguientes:

Paso 1: Definir la representación gráfica. Los elementos y conexiones que aceptará el DSL. Los elementos gráficos definidos no tienen una relación 1:1 con elementos del meta-modelo, pudiendo un mismo elemento gráfico tener diferentes significados dependiendo del contexto, o representar una serie de estructuras en el modelo.

Paso 2: Particularizar los estilos acorde a cada proyecto. Por cada representación se definieron los colores, íconos, formas de diagrama y etiquetas a mostrar, dando a cada una un significado semántico conocido por los expertos del dominio.

Paso 3: Definir las herramientas de edición y su comportamiento. Por cada representación se definieron las posibles acciones que es posible realizar dentro de la vista; éstas tendrán forma de ítems gráficos de una o más paletas de edición y de menú contextual para edición directa en el modelo. Una vez finalizada la especificación visual de las herramientas, se definió el comportamiento que tendrán, el cual se detalla utilizando OCL y Aceleo 3. Cada elemento dentro de la paleta puede crear, modificar y borrar múltiples instancias y conexiones de los elementos asociados al contexto en que se utiliza.

Paso 4: Validaciones. Por último se definieron un conjunto de validaciones que se aplicarán a cada representación. Una validación puede ser aplicada sobre los elementos de la representación gráfica o sobre el modelo. Las expresiones de validación se escriben utilizando OCL y Aceleo 3, verificando con el meta-modelo del DSL.

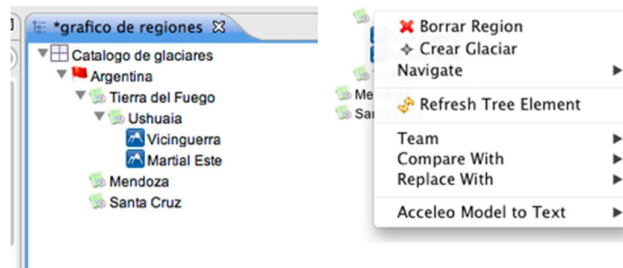


Fig. 3. Vista Catálogo de glaciares y Menú contextual

Vista Catálogo de glaciares. Es una vista jerárquica de los glaciares presentes en el modelo categorizados según el país, provincia y región a la que pertenece. Permite la creación y modificación rápida de glaciares dentro de una región para conformar un catálogo de glaciares. La Fig. 3 muestra un ejemplo de esta vista y el menú contextual

asociado a esta vista. Contiene las herramientas necesarias para crear o borrar instancias asociadas a los elementos gráficos, navegar el modelo y lanzar la generación de código del elemento marcado por el menú.

Diagrama de glaciares de una región. Describe los elementos que componen al conjunto de glaciares de estudio de una región y la forma en que se relacionan. Los elementos principales del meta-modelo tienen una representación gráfica en forma de ícono, sus relaciones están marcadas con diferentes colores dependiendo del contexto. La Fig. 4 muestra un listado de los elementos del meta-modelo y la representación que tienen en el diagrama:

	Glaciar		Modelo
	Estación de Medición		Gráfico
	Variable/Atributo simple		Transformada
	Proceso físico		Balance de masa
	Variable compuesta		Zona de Balance

Fig. 4. Elementos de la sintaxis concreta

Para conformar el modelo, los elementos se despliegan en un diagrama y son interconectados entre sí. El contenedor donde se despliegan los íconos representa una región. Tanto el elemento que representa una variable compuesta como el que representa el balance de masa se dibujan en el diagrama como un contenedor, que acepta, en el caso del balance de masa, zonas de balance y en el caso de una variable compuesta atributos simples y transformadas. La Fig. 5 muestra un Diagrama para el cálculo de balance de masa.

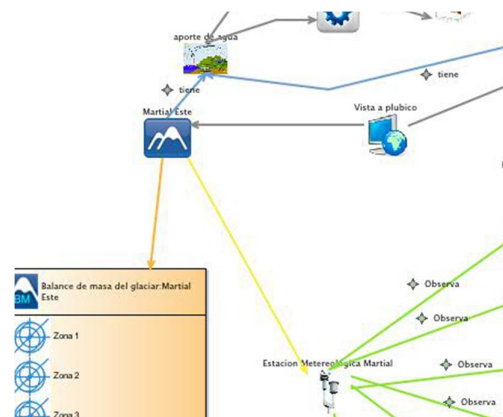


Fig. 5. Ejemplo de Diagrama de Glaciares con balance de masa

Los atributos de cada elemento del diagrama, (coordenadas de un glaciar, nombre y descripción), pueden ser editados desde la paleta de atributos derivada de la generación de código de EMF. Para la edición del diagrama se desarrolló una barra de herramientas. La paleta se encuentra dividida en 3 grupos: el principal, variables y presentación, como se muestra en la Fig. 6. La herramienta “Conector” verifica que los elementos interconectados tengan alguna relación en el meta-modelo, de lo contrario no permite unión de elementos.

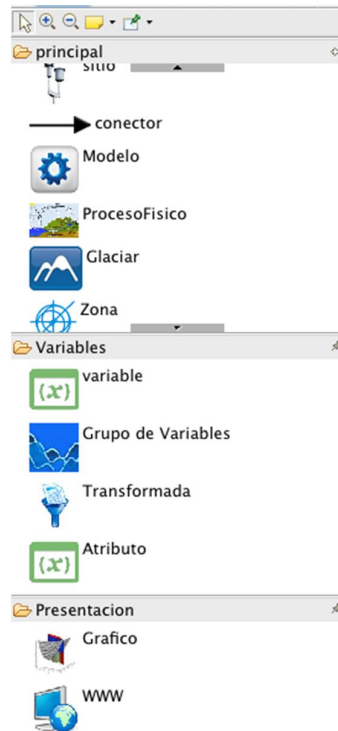


Fig. 6. Paleta de herramientas del Diagrama de glaciares

La validación del diagrama contiene un conjunto de sentencias OCL que verifican la correcta composición del modelo. Se verifica que se cumplan las cardinalidades mínimas de las relaciones descritas en el meta-modelo. De existir algún error se marca el elemento inconsistente y se muestra en la ventana de estado (Fig. 7).

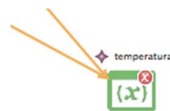


Fig. 7. Ejemplo de elemento en estado inválido

3.3 Generador de Código

Para la generación de código se realizó un plugin que explota la información contenida en el meta-modelo utilizando Aceleo 3 y OCL. Este plugin contiene las clases necesarias para configurar, disparar la generación y una serie de archivos templates que combinan el código a generar con consultas al modelo mediante el lenguaje, que contiene Aceleo 3, OCL y la información específica del meta-modelo.

3.4 Framework específico de Dominio

Se desarrolló un framework de dominio como interface entre la plataforma objeto y el generador de código. La aplicación generada corresponde a un sistema distribuido con componentes ejecutables en las siguientes plataformas: un servidor de aplicaciones JAVA, un servidor de cómputo estadístico R y clientes HTML5/Javascript. Los artefactos de software a generar se reducen al mínimo delegando responsabilidades en el framework a los fines de lograr que la calidad del código generada por la herramienta DSM sea de igual o superior a la del código escrito por un desarrollador.

4 Aplicación del DSL en el Balance de Masa de Glaciares

La validación del DSL y la herramienta DSM se realizó para dos glaciares de la región de Ushuaia: Martial y Vinciguerra. La Fig. 8 muestra el modelo asociado a la región de Ushuaia, donde se encuentran cargados los glaciares mencionados, sus estaciones meteorológicas, las variables medidas, las zonas de medición del balance de masa, los procesos con sus modelos asociados y un sitio informativo donde se muestran (en diferentes niveles) los datos de variables, balance de masa y caudal de agua aportado de ambos glaciares.

Puede observarse que cada glaciar tiene una estación meteorológica de toma de datos. El usuario experto modificó las propiedades asociadas a cada una de las variables: nombre, detalle, unidad de medición.

El principal proceso físico que se calcula es el balance de masa correspondiente a cateos sobre diferentes zonas del glaciar a lo largo de un periodo anual. Con la información recolectada se calcula el balance de masa total del glaciar. El DSL contiene una notación especial para el balance de masa de un glaciar. En el modelo se puede ver reflejada la medición del balance de masa de los dos glaciares. En ambos casos el experto seleccionó 10 zonas de cateos, de las cuales debió completar las propiedades: altitud, latitud, longitud, nombre, área. El otro proceso físico que puede observarse en el modelo, es el caudal de agua aportado por el glaciar.

La aplicación generada permite la gestión de datos de estudio, su análisis estadístico y publicación de resultados con el soporte del framework de dominio.

Para la gestión de datos de investigación la aplicación permite la carga de los datos específicos del balance de masa, las variables observadas y los procesos físicos. La Fig. 9 muestra una pantalla donde se observa la carga de los datos asociados al balance de masa del glaciar Martial. El investigador tiene la opción de cambiar el código R de los objetos que contengan funciones estadísticas: gráficos, procesos.

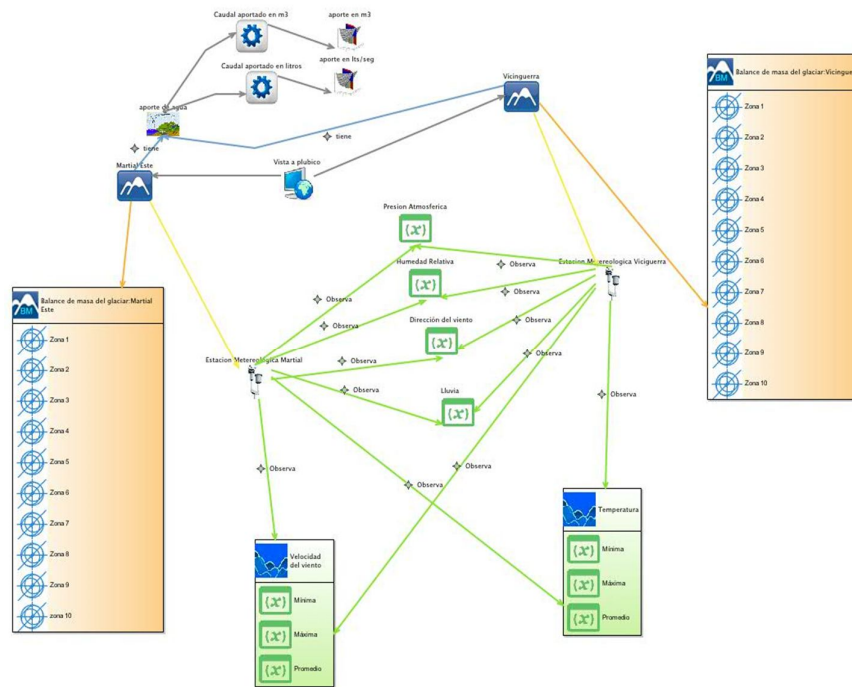


Fig. 8. Modelo de los glaciares Martial Este y Vinciguerra

Home Balance de Masa Variables Procesos Glaciar Martial Este 1 Campaña 2008

Zonas: Agregar fecha de medición, nueva Carga, Ver Cargas, Nueva Medición

Mediciones tomadas el día: 16-05-2008

zona	L	h	b	c	alfa	Bajo M. N. NL	Sale correp.	Delta H medido	Delta H adopt.	Día nieve hielo	Espesor m. muestr.	Esp. Nieve / Hielo	Dens.
zona 1													
zona 2								400					
zona 3								500					
zona 4								500					
zona 5													
zona 6													
zona 7													
zona 8													
zona 9													
zona 10													

zona 1 zona 2 zona 3 zona 4 zona 5

Universidad Nacional de Tierra del Fuego

Fig. 9. Pantalla de actualización de datos del Balance de Masa

La pantalla principal de la aplicación informativa muestra una imagen del glaciar con los objetos geo-localizables: zonas de balance, estaciones meteorológicas, desde la cual se puede acceder a los datos asociados.

5 Conclusiones

Este trabajo presenta un entorno DSM para aplicaciones de modelación glaciológica. Se muestra en particular el desarrollo de un DSL para estas aplicaciones, incluyendo una descripción de los conceptos y procesos centrales del dominio analizados con los expertos. Se describen la arquitectura y las fases de implementación del lenguaje y del generador que traduce los modelos definidos mediante el DSL en el código de las aplicaciones de los usuarios expertos. EL entorno DSM se completó con un Framework específico de dominio.

La notación del lenguaje es gráfica y sus distintos elementos resultan de un nivel de abstracción apropiado para los expertos que utilizan estas aplicaciones, presentando además gran flexibilidad para la construcción de modelos del comportamiento glaciario, mediante la adición de nuevos elementos en un modelo existente o mediante su intercambio. El DSL contempla además un entorno que facilita la integración de las distintas etapas del desarrollo de estos sistemas, desde la carga de datos hasta la presentación de resultados.

Si bien el dominio sobre el que se realizó el trabajo es de índole acotado y contempla los procesos principales del estudio de glaciares, el DSL puede extenderse fácilmente al cálculo y modelación de nuevos procesos en el dominio.

Referencias

1. Sonnessa M.: Modeling and Simulation of complex systems, PhD Thesis, Cultura e impresa, University of Torino, Italy, (2004)
2. Athanasiadis I., Villa F.: A roadmap to domain specific programming languages for environmental modeling. ACM, New York, 27-32, (2013)
3. Urciuolo A., Gel M., Iturraspe R., Moyano E., Villarreal M.: Lenguajes Específicos de Dominio para la Modelación de Ecosistemas Naturales. En: XVI WICC 2014, pp. 382-386 Ed. electrónica UNTDF, Ushuaia (2014)
4. Wal T. van der & van Elswijk M.J.B.: A generic framework for hydroenvironmental modelling. Proceedings HydroInformatics 2000, Ceder Rapids, IA, USA, (2000)
5. Günther O.: Environmental Information Systems. Springer-Verlag, Berlín, Germany, (1998)
6. Pohjonen, R., Kelly, S.: Domain-Specific Modeling, Dr. Dobb's Journal, August (2002)
7. Wegeler T., Gutzeit F., Destailleur A., Dock B.: Evaluating the benefits of using Domain-Specific Modeling Languages – an Experience Report. Proceedings of the 2013 ACM workshop on Domain-specific modeling, pp 7-12. ACM. New York, USA (2013)
8. Kelly, S., Tolvanen, J.: Domain-Specific Modeling Enabling Full Code Generation - IEEE Computer Society / John Wiley & Sons (2008)
9. Tolvanen, J., Kelly S.: Defining Domain-Specific Modeling Languages to Automate Product Derivation. Software Product Lines Vol. 3714, pp 198-209. Springer (2005)
10. Theisselmann F., Dransch D., Fischer J.: Model-Driven Development of Environmental Modeling Languages and Model Coupling. EnviroInfo 2009 (Berlín). Env. Informatics and Industrial Env. Protection Concepts, Methods and Tools. Shaker Verlag (2009)
11. Fowler M.: Domain-Specific Languages – USA - Addison-Wesley Professional (2011)
12. Gel M.: Modelado específico del dominio para aplicaciones de glaciología. Tesina de grado Lic. en Informática, UNPSJB Sede Ushuaia (2013).