

# Una evaluación por dimensiones para los lenguajes de transformación de modelos

## Evaluation using dimensions for model transformation languages

Juan C. Herrera R,<sup>1</sup> Francisca Losavio<sup>2</sup> & Alfredo J. Matteo L.<sup>2</sup>

1. Programa de Informática para la Gestión Social, Universidad Bolivariana de Venezuela
2. Laboratorio MoST, Escuela de Computación, Facultad de Ciencias, Universidad Central de Venezuela  
jchr1982@yahoo.com; {francislosavio@gmail.com, alfredo.matteo@ciens.ucv.ve}

Recibido para revisión 19 de abril de 2010, aceptado 04 de Octubre de 2010, versión final 29 de Octubre de 2010

**Resumen**— En el enfoque del desarrollo de software dirigido por modelos la utilización de un lenguaje de transformación de modelos juega un rol esencial. Es así, como el OMG (Object Management Group) a través de la especificación MOF (Meta-Object Facility) 2.0/QVT (Query/Views/Transformations) define un conjunto de lineamientos que debe ser satisfecho por un lenguaje de transformación de modelos. En estos últimos años muchos lenguajes han sido definidos y muchos trabajos sobre evaluación y/o comparación de estos lenguajes han sido publicados. Sin embargo, en cada una de estas comparaciones se evalúan aspectos bajo una sola perspectiva o dimensión. Este trabajo presenta una evaluación de lenguajes de transformación que agrupa los criterios y características bajo una cierta dimensión, como por ejemplo formalización, escenarios de uso, espacio tecnológico, arquitectura de ejecución, entre otros. Estas dimensiones, han sido identificadas a partir de trabajos realizados sobre comparaciones de lenguajes de transformación. El proceso de evaluación realizado se basa en la metodología DESMET con el enfoque de análisis de características, y el método para la toma de decisiones denominado AHP (Analytic Hierarchy Process). La consideración de varias dimensiones al mismo tiempo, facilita al desarrollador el proceso de selección de un lenguaje de transformación.

**Palabras Clave**— Desarrollo de software dirigido por modelos, lenguajes de transformación de modelos, MOF 2.0/QVT, DESMET, AHP.

**Abstract**—The approach of model driven software development requires the use of model transformation languages plays an essential role. The OMG (Object Management Group) through the MOF (Meta-Object Facility) 2.0/QVT (Query/Views/Transformations) specification defines a set of guidelines that must be satisfied by the processing language used. In recent years many languages have been defined and many works on the

evaluation and/or comparison of these languages have been published. However, in each of these comparisons, aspects are evaluated under a single perspective, such as scenarios of use, required technological space, and architecture among others. This work presents an assessment of transformation languages that groups the criteria and characteristics under certain dimensions. These have been identified from works carried out on comparisons of transformation languages. The proposed evaluation process is based on the DESMET methodology with analysis of characteristics approach, and the method of decision making called AHP (Analytic Hierarchy Process). The fact of considering multiple dimensions at the same time facilitates at developer in the selection process of a transformation language.

**Keywords**— Models driven software development, models transformation languages, DESMET, MOF 2.0/QVT, AHP.

### I. INTRODUCCIÓN

Es evidente que debido a los cambios continuos que experimentan las organizaciones, la creciente complejidad de los actuales sistemas y a su vez los rápidos avances tecnológicos en las plataformas que las implementan, la evolución de los sistemas de software es una tarea cotidiana para los desarrolladores, requiriendo un gran esfuerzo, que conlleva al incremento en los recursos que se destinan para ello. Como alternativa para facilitar la evolución de los sistemas de software surge el desarrollo de software dirigido por modelos (DSDM). Hecho evidenciado en OMG [22] y Pérez et al. [24] al indicar que a pesar del incremento en el nivel de abstracción en el área del desarrollo de software en los últimos años, aún quedan problemas por resolver, tales como la creciente complejidad de los sistemas y el surgimiento de nuevas plataformas. En consecuencia el

DSDM surge como paradigma que combina las siguientes familia de elementos: lenguajes específicos de dominio (DSL) que formalizan la estructura de unas aplicaciones particulares, el comportamiento y los requisitos en un dominio particular; descritos mediante metamodelos; y motores de transformación que analizan los modelos y generan el código fuente u otros modelos alternativos, acorde a reglas de transformación, permitiendo separar el conocimiento del dominio del negocio (modelo conceptual) de la plataforma tecnológica (modelo de implementación), así se reducirán los esfuerzos a la hora de la evolución de los sistemas.

El éxito del DSDM requiere de adecuados lenguajes de modelado y entre ellos los lenguajes de transformación de modelos (LTM) para escribir las reglas de transformación. En tal sentido Mens, Czarnecki y Van Gorp [20] han identificado requisitos de calidad que deben poseer estos lenguajes, tales como usabilidad, escalabilidad y conformidad a estándares. Además, marcos teóricos como el modelo de características discutido en [4] y la taxonomía de transformaciones de modelos presentada en [20] en alguna medida permiten comparar si la técnica de transformación utilizada por el LTM se acopla al propuesto por el OMG en la guía MDA [22], u otros enfoques utilizados en la práctica [5].

Muchos LTM son utilizados para especificar las definiciones de la transformación, en tal sentido, Kurtev [18] indica que aunque un lenguaje de propósito general (GPL) puede ser utilizado para escribir un programa que transforme un modelo en otro, la tendencia en la comunidad DSDM es la de utilizar DSL para escribir las definiciones de transformación. Esto se evidencia por el surgimiento de varias propuestas en respuesta a la solicitud realizada por el OMG [21] para la definición de un LTM bajo el estándar denominado QVT; entre ellas las más significativas son la del grupo QVT Merge [25] y QVT Partners [26]. Este hecho es confirmado por Czarnecki [6] al indicar que las transformaciones de modelos requieren de un DSL para su definición.

Por consiguiente, el establecer criterios adecuados para evaluar los LTM en el dominio del DSDM para la definición de transformaciones, es prioritario para la comunidad de desarrolladores que siguen este enfoque, por lo cual en esta investigación se identifican, seleccionan, unifican, agrupan y reubican los aspectos más relevantes a través de la definición de siete dimensiones vinculadas a las de formalización, organización, arquitectura, escenarios de uso, espacio tecnológico, aspectos no funcionales y herramientas de soporte. Esto es justificado por Van Gorp y Mens [30] al indicar que es conveniente establecer un conjunto de criterios concretos y adecuados que caractericen los LTM que ayude al desarrollador en la elección de un LTM, acorde a necesidades específicas.

Para realizar una comparación bajo la perspectiva de estas dimensiones, se ha definido un método basado en la metodología DESMET [17] como mecanismo que permite planificar y ejecutar un ejercicio de evaluación que sea lo más imparcial y

confiable. El método utiliza el enfoque de análisis de características para realizar la evaluación.

Con la finalidad de fortalecer y ampliar los resultados técnicos obtenidos al analizar los LTM a través de sus dimensiones, es conveniente cotejar y comparar las puntuaciones obtenidas con la finalidad de establecer el orden relativo de los méritos de los LTM, tomando en cuenta aspectos cualitativos de las preferencias encontradas en los trabajos analizados; en este sentido el Proceso Analítico Jerárquico (Analytic Hierarchy Process, AHP) [29] es utilizado como método que descompone una situación compleja, la evalúa y entrega una lista ordenada de las alternativas de solución desde la mejor hasta la peor.

Este artículo se ha estructurado en tres secciones; además de la introducción, la sección 2 establece la metodología de evaluación y finalmente la sección 3 con las conclusiones.

## II. METODOLOGÍA DE EVALUACIÓN

En esta investigación, se sigue el enfoque sistémico, propuesto por Kitchenham [17] denominado DESMET; esta metodología ayuda a un evaluador en una organización a planificar y ejecutar un proceso de evaluación que es imparcial y confiable con el objetivo de identificar el método de evaluación más adecuado para un producto o elemento de software que se adapte a un contexto específico. La metodología propone varios métodos de evaluación y los caracteriza estableciendo criterios técnicos y prácticos para seleccionar el método de evaluación más adecuado. Para la evaluación de los LTM, se escogió el método de evaluación denominado análisis de características por selección o filtrado. Este método es utilizado para describir una evaluación cualitativa y estructurada permitiendo caracterizar los LTM para poder compararlos, y con ello facilitar la toma decisión a la hora de elegir los lenguajes que se adecuen a las necesidades del desarrollador.

Los criterios técnicos definidos por la metodología DESMET que permitieron determinar que el método de análisis de característica es el adecuado para la evaluación de los LTM son:

- Contexto de evaluación: se han identificado los requerimientos que los usuarios tienen para una particular actividad en el DSDM y están relacionadas a características que un lenguaje aspira, para soportarla; particularmente, implica la preselección inicial de muchos lenguajes candidatos alternativos que posteriormente serán evaluados con más detalle.
- Naturaleza del impacto: evaluación cualitativa de los LTM.
- Naturaleza del objeto de evaluación: se evalúa los LTM dentro del paradigma del DSDM.
- Alcance del impacto: se consideran solamente las dimensiones individuales de los LTM: formalización, organización, escenarios de usos, arquitectura, espacio tecnológico, aspectos

no funcionales y herramientas de soporte.

- Madurez del LTM: los LTM analizados disponen de la suficiente información para el enfoque cualitativo.
- Tiempo de aprendizaje: se dispone del tiempo necesario para comprender los principios subyacentes de los LTM.
- Evaluación de la madurez de la organización: no existe una organización que hará la evaluación, por lo cual, será ejecutada por una persona, responsable de todo el proceso, donde la información obtenida se basa principalmente en la documentación y se utiliza frecuentemente como una primera etapa en un ejercicio de evaluación más complejo que implica la reducción a uno o dos posibles candidatos.

A continuación se describe en detalle la instanciación del método de análisis de características por selección para la evaluación de los LTM:

### **2.1. Método de Análisis de Características por selección o filtrado**

La implementación del método fue organizada en seis pasos, su especificación se indica a continuación, según Kitchenham [17]:

**Tabla 1.** LTM a ser evaluados agrupados según la técnica de implementación utilizada [5].

Técnica de implementación	Descripción	LTM
1 Manipulación Directa	Permite la representación interna del modelo y APIs para manipularlo.	JMI [14]
2 Dirigido por la Estructura	Crea la estructura jerárquica del modelo, luego establece los atributos y referencias en él.	OptimalJ [9]
3 Operacional	Extiende el formalismo de un metamodelo mediante librerías/frameworks para expresar cálculos/operaciones.	KERMETA [31] SmartQVT [8]
4 Relacional	Establece relaciones entre elementos origen y destino utilizando restricciones.	KMTL [2] MOMENT [27]
5 Transformación de Grafos	Representa gráficamente elementos del modelo facilitando la especificación y comprensión de las transformaciones.	GReAT [1]
6 Híbrido	Combina técnicas anteriores.	ATL [3]
7 Meta-Programación	Utiliza un DSL embebido en un lenguaje procedural.	RubyTL [7]
8 XSLT	Transformación definida utilizando el lenguaje XSLT.	UMT [12]

### **2.1.2 Definir las propiedades o características para la evaluación del conjunto de LTM seleccionados**

Una vez identificado los LTM a evaluar, se definen las características que se utilizan para realizar la evaluación. Para ello se realizan las siguientes actividades:

- **Recolección de información:** Esta actividad involucra el estudio de las características relevantes en los LTM obtenidas de la literatura relacionada con las técnicas y lenguajes utilizados para la transformación de modelos, incluyendo especificaciones formales como QVT [23], para la definición de los LTM. Para la selección de los criterios o propiedades de evaluación que conforman la propuesta se realizó un compilado de los criterios utilizados en las investigaciones que sirvieron de antecedentes en este trabajo: [1], [3], [9], [10], [12], [16], [19], [20], [23], [27], [28] y [30].

#### **2.1.1 Seleccionar un conjunto de lenguajes candidatos para evaluar.**

De acuerdo al estudio de los trabajos recopilados en la investigación, se han identificado veintisiete (27) LTM. Debido a la gran cantidad de lenguajes, se hace necesario elegir cuales son los más relevantes a fin de reducir el conjunto de lenguajes a evaluar. Para la selección de los LTM se utilizaron los siguientes criterios: el primer criterio fue ubicar los LTM según la técnica de implementación (ver tabla 1) utilizada en el contexto del DSDM descrito por Czarnecki y Helsen [5]. El segundo criterio fue seleccionar representaciones de al menos uno y máximo dos LTM representativos por cada técnica en base al crecimiento, aceptación que han tenido en la comunidad DSDM, el cual es reflejado en función de la cantidad trabajos publicados y la utilización de estos en la solución de problemas a través de herramientas que soportan a los lenguajes. Finalmente, en la tabla 1 se muestran los resultados del proceso de selección, indicando en la columna 3 los LTM que serán evaluados.

- **Identificación de dimensiones:** Al examinar diversos trabajos donde existen comparaciones, análisis de características y propuestas de requisitos funcionales y no funcionales en los LTM, se han identificado que existen diferentes ámbitos o dimensiones en las cuales se sitúan estos trabajos. Así por ejemplo existen trabajos tales como [10], [23] en los cuales el análisis realizado se fundamenta en lo relativo a como los lenguajes son definidos; también existen otros trabajos como [16], [28] que solo consideran como están organizados estos lenguajes. Es importante resaltar que el método de análisis de características por selección se basa en identificar los requisitos que los desarrolladores tienen por actividad o tarea particular, en este caso la transformación de modelos y relaciona esas necesidades con características o funciones que el LTM debe apoyar para realizarla, es decir: que el LTM debe estar formalmente definido, que permita diversas formas

de organizar los elementos del lenguaje, que facilite la especificación de transformaciones para diferentes escenarios de uso, entre otros, en consecuencia las características identificadas fueron agrupadas en cada una de las dimensiones que vienen a representar los requisitos anteriormente descritos. Finalmente, de ahí se han identificado siete dimensiones: *formalización, organización, arquitectura, escenarios de uso, espacio tecnológico, aspectos no funcionales del lenguaje y herramienta*; cada una representando

una macro categoría para la evaluación de los LTM. En la figura 1 se muestra el modelo conceptual de un LTM que indica que un lenguaje es implementado a través de una o varias técnicas y que tiene asociado unas características que poseen atributos que sirven como indicadores para evaluar el lenguaje, donde cada característica posee dos atributos, el primero, indica la *dimensión* a la que pertenece la característica; el segundo, indica si la característica es *obligatoria*.

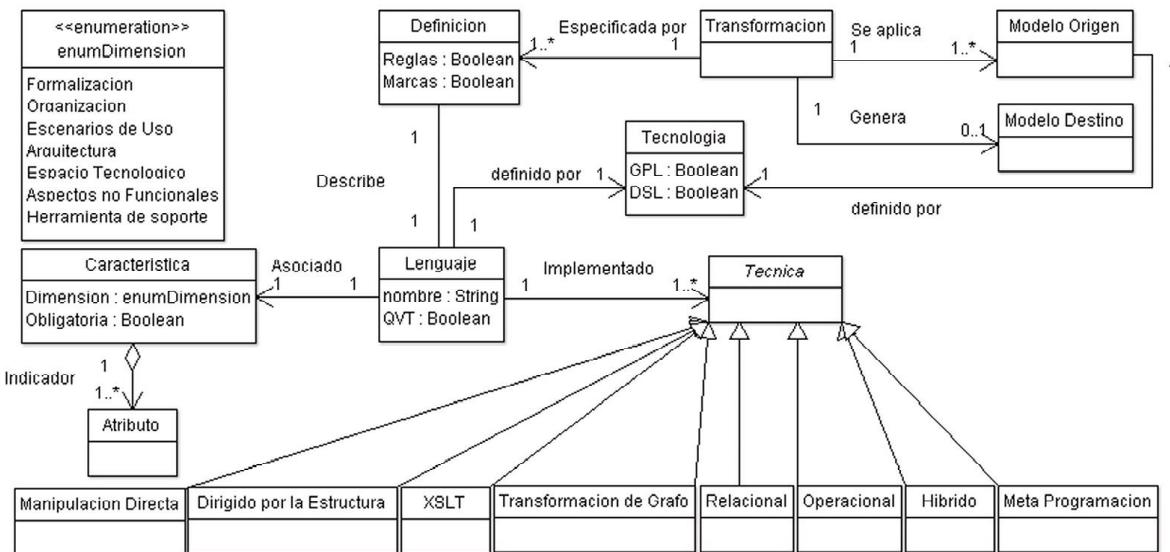


Figura 1. Modelo conceptual de un LTM

A continuación se describe el significado de cada una de las dimensiones:

- 1. Formalización del Lenguaje:** Agrupa las características que un LTM debe satisfacer para proveer un soporte formal para la especificación y/o descripción de las transformaciones de modelo.
- 2. Organización del Lenguaje:** Agrupa las características que representan las formas en la cual las sentencias, reglas del lenguaje pueden ser organizadas.
- 3. Arquitectura:** Agrupa las características que describen los mecanismos que dispone un LTM para que pueda ejecutar sus funcionalidades, es decir la implementación subyacente.
- 4. Escenarios de Uso:** Agrupa las características que describen los posibles escenarios en que pueden aplicarse los LTM. Desde el punto de vista de lo que el lenguaje puede hacer.
- 5. Espacio Tecnológico:** Agrupa las características basadas en el concepto de espacio tecnológico definido en [15] como un contexto de trabajo que posee un conjunto de conceptos relacionados; agrupa el conjunto de tecnologías basadas en estándares asociados a los LTM.
- 6. Aspectos no Funcionales:** Agrupa las características de calidad deseables en los LTM desde el punto de vista de los usuarios finales.
- 7. Herramienta de soporte:** Agrupa las características sobre

los entornos de desarrollo orientados al DSDM que implementan los LTM y que son de gran importancia al momento de elegir un lenguaje.

- Distribución de características:** En este artículo se propone el siguiente procedimiento para el desarrollo de esta etapa. *Primero*, para cada uno de los trabajos examinados se sitúan las características identificadas por los autores con respecto a las dimensiones previamente definidas. *Segundo*, se procede a la **aplicación de un proceso cognitivo** implementado a través del uso de referencias cruzadas entre las características en su correspondiente dimensión; donde los siguientes criterios fueron utilizados: **Unificación**, agrupa características cuyas semánticas son consideradas iguales, incluye agrupar y unificar atributos. **Conversión**, transforma características en atributos y las agrupa dentro de una característica nueva o existente. **Incorporación**, anexa nuevas características, producto de la investigación. **Reubicación**, existen trabajos que abarcan diferentes dimensiones, por lo cual algunas características deben ser reubicadas. Como resultado, se generó una única tabla, para cada una de las siete (7) dimensiones, con las correspondientes características y sus atributos de medición identificados en todos los trabajos analizados (ver tablas 2, 3 y 4); las características consideradas por los autores como obligatorias se identifican con un asterisco (\*) después del nombre de la característica.

**Tabla 2.** Características para la evaluación de los LTM en la dimensión *escenarios de uso*.

Características	Atributos	Descripción
Traducción*	Migración	Un modelo es transformado en otro en el mismo nivel de abstracción.
	Síntesis	Un modelo se transforma en otro a un nivel inferior de abstracción.
	Ingeniería inversa	Un modelo es transformado en otro a un nivel superior de abstracción.
	Endógeno	Todos los modelos son conforme al mismo metamodelo.
	Exógeno	Múltiples metamodelos son utilizados.
	Reemplaza	Modelo de entrada sí se convierte en modelo de salida.
Interpretación	Crea	Modelo de entrada: solo lectura. Modelo de salida: se genera.
	Normalización	Un modelo es transformado por reducirlo a un sub lenguaje.
	Refactorización	Un modelo es reestructurado, mejorando su diseño.
	Corrección	Un modelo es modificado a fin de fijar un error.
Sincronización*	Adaptación	Un modelo es modificado a fin de adaptarse a nuevos requerimientos.
	Parcial	Solo aplica a determinadas partes o elementos de un modelo.
	Comprobación	Verifica que los modelos estén relacionados de una forma determinada.
	Doble sentido	Genera el modelo a partir del código fuente (LTM) y viceversa.
Orden superior	Vistas	Un modelo completamente derivado de otro modelo. Un solo sentido.
		Permite aplicar una transformación a otra transformación (modelo).

**Tabla 3.** Características para la evaluación de los LTM en la dimensión *organización*.

Características	Atributos	Descripción
Modularidad a nivel de reglas*	Composición	Permite el desarrollo de transformaciones mediante módulos.
	Combinarse	Permite combinar módulos entre sí para producir otros módulos.
Modularidad a nivel de Transformación	Encadenarse	Permite encadenar varias transformaciones consecutivamente.
	Composición	Permite componer transformaciones para formar una nueva.
Definición de Condiciones de ejecución*	Combinarse	Permite combinar varias transformaciones.
	Precondición	La aplicación de las reglas es condicionada a algún contexto.
	Determinístico	Un algoritmo dirige el orden de ejecución de las reglas.
	Iterativo	Provee estructuras de iteración de reglas, o mecanismos de recursión.
	No determinístico	No hay orden de ejecución predefinido entre las reglas.
Mecanismos para reutilizar e integrar	Interactivo	Permite al usuario especificar la estrategia de ejecución de la regla.
	Importando	Importando librerías de transformación.
	Por parámetros	Permite especializar transformaciones utilizando parámetros.
	Deshabilitar	Permite restringir el conjunto de objetos al cual aplica una regla.
Correspondencia de elementos*	Reemplazo	Permite la sobreescritura de reglas.
	Herencia	Permite construir nuevas reglas a partir de reglas existentes.
	Colección	Permite explícitamente iterar sobre los elementos de una colección.
	Simple	Expresado en términos de un simple elemento.
Definición de elementos	Por elementos	Múltiples elementos destino pueden ser definidos en una sola regla.
	Por reglas	Múltiples reglas proveen valores de propiedad al mismo objeto destino.
Transformaciones Intermedias		Permite la definición de transformaciones intermedias, apoyando así las transformaciones multipaso.

**Tabla 4.** Características para la evaluación de los LTM en la dimensión *formalización del lenguaje*

Características	Atributos	Descripción
Notación Sintaxis Abstracta*	Textual	Notación textual.
	Metamodelo	Un único metamodelo para su representación.
	Composición de metamodelos	Un metamodelo puede ser la composición de varios metamodelos.
Notación Sintaxis Concreta*	Textual	Permite definir transformaciones sin una herramienta gráfica.
	Partes	Solamente partes de una transformación pueden ser graficadas.
	Simple	Una transformación sencilla puede ser completamente definida.
	Composición	Permite componer varias transformaciones.
Estilo de la Definición	Declarativo	Describe relaciones entre variables en términos de funciones o reglas.
	Imperativo	Describe explícitamente los pasos a seguir para producir un resultado.
Expresión (Conjunto de Constructos)	Conciso	El lenguaje tiene pocas construcciones sintácticas.
	Verboso	El lenguaje permite introducir extras sintácticos.
	General	Grupo de general construcciones.
	Especializado	Especializadas construcciones.
Representación Semántica del lenguaje	Enfoque denotacional	Es definida por una función que la relaciona con una estructura matemática.
	Enfoque operacional	Es definida mediante una máquina abstracta y las traducciones constituyen el metalenguaje que define las semánticas del lenguaje.
Cuerpo reglas de transformación	Separación sintáctica	Diferencia las partes de una regla de transformación que operan sobre el modelo origen de las partes que operan sobre el modelo destino.
	Estructuras intermedias	El lenguaje recurre a alguna estructura adicional para describir la transformación, que no es parte del modelo a transformar.
	Parametrización	Permite el uso de parámetros de control que permiten el paso de valores.

Características	Atributos	Descripción
Correspondencia de Elementos*	Patrón utilizado por el lenguaje para especificar el origen de una regla	Patrón utilizado por el lenguaje para especificar el origen de una regla
	Por colección	Correspondencia de conjuntos de elementos del modelo origen.
	Por tipo	Selección de elementos por tipo-preciso, incluye instancias de sub-tipos.
	Por tipo-preciso	Selección de elementos por tipo-preciso, excluye instancias de sub-tipos.
	Filtrado condicional	Filtrar el conjunto de coincidentes registros o tuplas sobre la base de asociaciones, valores de atributo, y otros criterios (condiciones).
	Filtrado recursivo	Manejo recursivo de la estructura con niveles arbitrarios de anidamiento.
	Tipo de reglas expresadas para definir la relación entre entidades.	Tipo de reglas expresadas para definir la relación entre entidades.
	Reglas sobre clases	Especifica como las clases en el modelo destino son instanciadas.
Orientación	Reglas sobre atributos	Especifica como los valores de los atributos son obtenidos.
	Orientado a Objeto	La regla de transformación está estructurada alrededor de objetos.
Notaciones	Orientado a Aspectos	La regla de transformación está estructurada alrededor de conceptos semánticos; ejem: reemplazar una nomenclatura de un sistema con otra.
	Aplicadas a un dominio específico	Uso de notaciones existentes utilizadas por el lenguaje para representar las transformaciones. Ejemplo: basado en MOF.
Ámbito del lenguaje	DSL	Un LTM, es un DSL que captura todas las características del dominio de la transformación de modelo (navegación, consulta y modificación, etc.).
	GPL	Las reglas y la planificación son implementadas desde cero, utilizando un lenguaje de programación general.
Homoiconicity		El lenguaje utilizado para expresar un modelo también está relacionado o es idéntico al lenguaje de transformación.
Propiedades matemáticas		Si el lenguaje de transformación tiene un fundamento matemático.

### **2.1.3 Asignar prioridades a las características por cada dimensión tomando en cuenta las necesidades o requisitos del usuario**

Cada característica está acompañada por un grado de estimación de importancia. Un buen LTM es aquel que incluye las características más importantes para los usuarios. Su importancia puede ser estimada al considerar si esta es obligatoria o deseable. Para valorar una característica la siguiente escala ordinal es considerada: Obligatorio, Deseable; la asignación de la prioridad está estrechamente vinculada con los requisitos obligatorios y opcionales establecidas en los trabajos analizados (ver tablas 2, 3 y 4).

#### **2.1.4 Convenir a un sistema de puntuación que puede aplicarse a todas las características**

La evaluación utiliza la escala nominal que permite evaluar características simples, donde se detecta la presencia o ausencia de la característica en el LTM. La presencia es identificada por el signo (+), la ausencia carece de tener signo. Cada característica tiene sus atributos que actúan como indicadores que determinan si el lenguaje presenta la característica. El valor final será obtenido tomando en cuenta los siguientes criterios: primero, por la sumatoria de todas las características que satisfaga el lenguaje; segundo, por la sumatoria de todos los atributos presentes; tercero, se verifica que todas las características obligatorias estén presentes en el lenguaje. En consecuencia,

el LTM con mayor puntuación es aquel que posee el mayor número de características, si hay dos lenguajes con igual cantidad de características satisfechas, se tomará en consideración el número de atributos presentes, indicando que la característica está mejor implementada. Un LTM que satisfaga todas las características obligatorias, entonces satisface la dimensión que las agrupa. Por otro lado, es inaceptable un LTM que no posea alguna característica obligatoria.

**2.1.5 Llevar a cabo la evaluación para determinar que tan bien los LTM satisfacen los criterios previamente establecidos**

La evaluación es llevada a cabo al analizar los ejemplos propuestos en cada uno de los trabajos identificados que describen las características y funcionalidad del lenguaje además de esbozar las fortalezas y debilidades presentes, contrastándolas con las características identificadas y agrupadas según la dimensión a la cual pertenecen. Los resultados de la evaluación son presentados en forma de tabla de dos ejes (ver tablas 5, 6 y 7). En el eje vertical están las características y sus atributos por dimensión (ver tablas 2, 3 y 4), y en el eje horizontal están los LTM evaluados (ver tabla 1). Debido a limitaciones de espacio, solo se muestran 3 de las 7 dimensiones, los detalles de las otras dimensiones pueden verse en [13]. Esto permitirá al evaluador comparar los LTM desde la perspectiva de la dimensión de interés.

**Tabla 5.** Resultados de la evaluación de los LTM para la dimensión *escenarios de uso*.

Características	Atributos	Lenguajes									
		JMI	OptimalJ	Kermeta	SmartQVT	KMTL	MOMENT	GReAT	ATL	RubyTL	UMT
	Exógeno				+		+	+	+		
	Reemplaza									+	
	Crea	+				+		+	+		
	Normalización										
Interpretación	Refactorización			+						+	
	Corrección										
	Adaptación				+		+				
	Parcial	+									
Sincronización *	Comprobación	+			+		+				
	Doble sentido	+		+		+		+	+	+	
	Vistas	+			+		+				
Orden superior										+	
	<b>Totales</b>	2(2)	2(6)	3(3)	3(6)	2(3)	3(6)	2(4)	4(8)	2(2)	2(4)

+ = satisface, \* = requisito obligatorio

**Tabla 6.** Resultados de la evaluación de los LTM para la dimensión organización.

Características	Atributos	Lenguajes									
		JMI	OptimalJ	Kermeta	SmartQVT	KMTL	MOMENT	GReAT	ATL	RubyTL	UMT
Modularidad *	Composición	+	+	+	+	+	+	+	+	+	+
	Combinarse										+
Composición	Encadenarse										
	Componerse			+			+				+
	Combinarse										
Definición de condiciones de ejecución *	Precondición	+		+		+	+	+	+		
	Determinístico	+	+	+	+	+	+	+	+	+	+
	Iterativo										+
	No determinístico								+	+	
	Interactivo										
Mecanismos para reutilizar e integrar	Importando	+		+						+	+
	Por parámetros		+	+							
	Deshabilitar										
	Reemplazo			+							+
	Herencia			+	+						+
Definición de elementos destino	Múltiples elementos por una sola regla										+
	Múltiples reglas para un solo elemento	+									+
Transformaciones	Intermedias			+							+
Direccionalidad*	Unidireccional	+	+	+	+		+	+	+	+	+
	Bidireccional					+					
Cardinalidad	Muchos a muchos										
	Uno a uno	+	+	+	+	+	+	+	+	+	+
	Muchos a uno										
	Uno a muchos										
	<b>Totales</b>	6(7)	5(5)	7(11)	5(5)	4(5)	5(6)	5(8)	5(7)	8(11)	3(3)

+ = satisface, \* = requisito obligatorio

**Tabla 7.** Cuadro de resultados de la evaluación de los LTM para la dimensión *fórmalización del lenguaje*.

Características	Atributos	Lenguajes								
		JMI	OptimalJ	Kermeta	SmartQVT	KMTL	MOMENT	GReAT	ATL	RubyTL
Notación Sintaxis Abstracta*	Textual			+	+		+		+	
	Metamodelo			+	+		+		+	
	G Composición de Metamodelos									
Notación Sintaxis Concreta*	Textual	+	+	+	+	+	+	+	+	+
	Partes									+
	G Simple									+
	Composición									
Estilo de la Definición	Declarativo					+			+	+
	Imperativo	+	+	+	+	+	+		+	+
	Conciso	+				+				
Expresión (Conjunto de Constructos)	Verboso	+	+	+	+		+	+	+	+
	General					+				+
	Especializado	+	+							
Representación Semántica	Enfoque denotacional			+						+
	Enfoque operacional			+						

Características	Atributos	Lenguajes									
		JMI	OptimalJ	Kermeta	SmartQVT	KMTL	MOMENT	GReAT	ATL	RubyTL	UMT
Cuerpo reglas de transformación	Separación sintáctica							+			
	Estructuras intermedias							+			
	Parametrización										
	Por colección	+		+	+	+	+		+		+
	Por tipo	+	+	+	+	+	+	+	+	+	+
	Por tipo-preciso	+									
Correspondencia de elementos*	Filtrado condicional									+	+
	Filtrado recursivo									+	+
	Reglas sobre clases										
	Reglas sobre atributos										
Orientación	Orientado a Objeto	+	+	+			+				
	Orientado a Aspectos										
Notaciones	Aplicadas a un dominio específico			+							
Ámbito del Lenguaje	DSL		+	+	+	+	+	+	+	+	+
Homoiconicity	GPL		+								
Propiedades	Matemáticas										
<b>Total (15)</b>		6(9)	7(8)	7(12)	6(8)	6(9)	6(8)	6(10)	6(7)	5(9)	5(9)

+ = satisface, \* = requisito obligatorio

### 2.1.6 Analizar e interpretar resultados

Para realizar la interpretación de resultados, se ha utilizado el método propuesto en la década de los 80 por el matemático Thomas L. Saaty, denominado Proceso Analítico Jerárquico (Analytic Hierarchy Process, AHP) [29], tomando en cuenta los resultados obtenidos en la sección 2.1.5. AHP es un método formal de estimación basado en múltiples criterios de decisión, utilizado cuando se tienen que considerar aspectos tanto cuantitativos como cualitativos en una decisión; que da como

resultado una jerarquización de las opciones que se expresa en un número único asignado a cada opción. AHP es un método para la toma de decisiones donde hay un número limitado de alternativas, donde cada una tiene un número de diferentes criterios, algunos de los cuales pueden ser difíciles de formalizar. AHP ayuda con la identificación y ponderación de criterios de selección, analizando los datos recolectados para el criterio, y agiliza el proceso de toma de decisión. Para ello las puntuaciones obtenidas deben ser cotejadas y comparadas para decidir el orden relativo de los méritos de las alternativas o LTM.

**Tabla 8.** Escala fundamental para comparaciones en pareja o grupo de dos<sup>1</sup>

Cuando el criterio a(i) es comparado con el criterio a(j), la:		
Intensidad	Definición	Explicación
1	Igualmente importante	Dos criterios contribuyen igualmente para el objetivo
2	Débil o Leve	
3	Ligeramente más importante	La experiencia y juicio ligeramente favorece un criterio sobre el otro
4	Moderadamente más importante	
5	Notablemente más importante	La experiencia y juicio fuertemente favorece un criterio sobre el otro
6	Fuertemente más importante	
7	Demostrablemente más importante	La experiencia y juicio muy fuertemente favorece un criterio sobre el otro. Su predominio es demostrado en la práctica.
8	Muy , muy fuerte	
9	Absolutamente más importante	La evidencia favorece a un criterio sobre el otro contundentemente
Recíprocos de los anteriores	Si la actividad i tiene asignado uno de los valores anteriores cuando este se compara con la actividad j, entonces j tiene el valor recíproco cuando se compara con i	Una suposición razonable o lógica
1.1 – 1.9	Si las actividades son muy cercanas	La importancia relativa entre las actividades es mínima, pero importante de señalar.

Para realizar las *comparaciones*, se necesita una escala de números (ver tabla 8) que indique cuantas veces es más importante un elemento sobre otro elemento con respecto al criterio al cual ellos son comparados. Las *prioridades* son números asociados a los elementos de una jerarquía AHP. Ellos representan los pesos relativos de los elementos en cualquier

grupo. Al igual que las probabilidades, las prioridades son cifras absolutas entre cero y uno. Dependiendo del problema, el “peso” puede referirse a la importancia, preferencia, probabilidad, o cualquier otro factor que esté siendo considerado. El procedimiento utilizado en este trabajo para el análisis de resultados aplicando el método AHP consta de los siguientes pasos: (a) Establecer el objetivo general del análisis; (b) Seleccionar los criterios de evaluación; (c) Seleccionar las alternativas a evaluar; (d) Estimar los criterios de comparación;

1. Escala diseñada por el Dr. en Matemáticas Thomas Saaty (Matemático de la Universidad de Pennsylvania).

(e) Comparar por pares las alternativas desde la perspectiva de cada criterio; (f) Colocar la puntuación relativa para cada alternativa a evaluar por cada una de los criterios para la determinación de los pesos locales; (g) Obtener la puntuación relativa por cada criterio para la determinación de los pesos globales.

#### 2.1.6.1 Resultados de la aplicación del método AHP

- La estructura del modelo jerárquico que establece el objetivo a alcanzar, la selección de los criterios (**dimensiones**) de evaluación, y la selección de las alternativas (**LTM**) a evaluar son especificadas en la tabla 9.

**Tabla 9.** Modelo jerárquico AHP

Objetivo	Elegir el mejor LTM para el DSDM.
Dimensiones	Formalización, Organización, Espacio tecnológico, Escenarios de uso, Aspectos no funcionales, Arquitectura y Herramienta de soporte.
LTM	JMI, OptimalJ, KERMETA, SmartQVT, KMEL, MOMENT, GReAT, ATL, RubyTL y UMT.

- Estimaciones acerca de todos los criterios de comparación (importancia relativa de cada una de las dimensiones en términos de su contribución al logro del objetivo)

Se basa en las preferencias subjetivas que hacen los autores a partir del análisis de los trabajos que tratan sobre los LTM, es decir, se establecen las prioridades tomando en consideración los datos recabados en la investigación y procesados matemáticamente. Se toma en cuenta la importancia relativa entre

las dimensiones, por ejemplo, para algunos autores la dimensión formalización del lenguaje es más importante que la dimensión organización, y así sucesivamente. La valoración sobre la importancia de la dimensión es expresada en términos cualitativos y mediante una escala previamente establecida por el método AHP (ver tabla 8) de la que se obtienen los valores numéricos que se corresponden con su valoración. La tabla 10 muestra las prioridades obtenidas entre las dimensiones donde se establece la importancia relativa que el decisor otorga a cada dimensión.

**Tabla 10.** Preferencias sobre las Dimensiones

Dimensión	Formalización	Organización	Escenarios	Espacio	Aspectos	Arquitectura	Herramienta	Prioridad
Formalización	1	2	2	3	3	3	5	<b>0,287</b>
Organización	1/2	1	2	2	2	2	2	<b>0,181</b>
Escenarios	1/2	1/2	1	3	3	3	3	<b>0,189</b>
Espacio	1/3	1/2	1/3	1	2	2	3	<b>0,116</b>
Aspectos	1/3	1/2	1/3	1/2	1	2	3	<b>0,097</b>
Arquitectura	1/3	1/2	1/3	1/2	1/2	1	3	<b>0,080</b>
Herramienta	1/5	1/2	1/3	1/3	1/3	1/3	1	<b>0,049</b>
	3,20	5,50	6,33	10,33	11,83	13,33	20,00	

**Tabla 11.** Test de consistencia de los juicios emitidos sobre las preferencias en las dimensiones

Test de Consistencia		
$\lambda =$	7,512146305	$IA = 1,341$
$IC =$	0,085357718	como $RC \leq 0,10$
$RC =$	<b>0,063652288</b>	entonces los juicios son consistentes

En la tabla 11 se muestra los resultados del test de consistencia para comprobar a través del radio de consistencia (RC) que los juicios del tomador de decisión reflejada en la tabla 10 son consistentes.

- Comparación por pares de LTM desde la perspectiva de cada una de las Dimensiones.

Se comparan por pares de LTM (JMI, OptimalJ, Kermeta, SmartQVT, KMEL, MOMENT, GREAT, ATL, RubyTL, UMT)

desde la perspectiva de cada Dimensión (Formalización, Organización, Escenarios de uso, Espacio tecnológico, Aspectos no funcionales, Arquitectura y Herramienta de soporte) tomando en cuenta los valores obtenidos de la aplicación del método de análisis de características de la metodología DESMET (ver tablas 5, 6 y 7). En la tabla 12 se muestra los resultados obtenidos de la comparación por pares de LTM solo con respecto a la dimensión formalización debido a limitaciones de espacio, sin embargo el procedimiento se realizó de forma similar para las otras dimensiones.

**Tabla 12.** Comparación por pares de LTM para la Dimensión Formalización.

A	B	LTM		Valores		Más Importante	Intensidad
		A	B	A	B		
JMI	OPTIMALJ	6(9)	7(8)	B		3	
JMI	KERMETA	6(9)	7(12)	B		4	
JMI	SMARTQVT	6(9)	6(8)	A		2	
JMI	KMTL	6(9)	6(9)	A		1	
JMI	MOMENT	6(9)	6(8)	A		2	
JMI	GREAT	6(9)	6(10)	B		2	
JMI	ATL	6(9)	6(7)	A		2	
JMI	RUBYTL	6(9)	5(9)	A		3	
JMI	UMT	6(9)	5(9)	A		3	
OPTIMALJ	KERMETA	7(8)	7(12)	B		2	
OPTIMALJ	SMARTQVT	7(8)	6(8)	A		3	
OPTIMALJ	KMTL	7(8)	6(9)	A		3	
OPTIMALJ	MOMENT	7(8)	6(8)	A		3	
OPTIMALJ	GREAT	7(8)	6(10)	A		3	
OPTIMALJ	ATL	7(8)	6(7)	A		3	
OPTIMALJ	RUBYTL	7(8)	5(9)	A		5	
OPTIMALJ	UMT	7(8)	5(9)	A		5	
KERMETA	SMARTQVT	7(12)	6(8)	A		4	
KERMETA	KMTL	7(12)	6(9)	A		4	
KERMETA	MOMENT	7(12)	6(8)	A		4	
KERMETA	GREAT	7(12)	6(10)	A		3	
KERMETA	ATL	7(12)	6(7)	A		4	
KERMETA	RUBYTL	7(12)	5(9)	A		5	
KERMETA	UMT	7(12)	5(9)	A		5	
SMARTQVT	KMTL	6(8)	6(9)	B		2	
SMARTQVT	MOMENT	6(8)	6(8)	A		1	
SMARTQVT	GREAT	6(8)	6(10)	B		2	
SMARTQVT	ATL	6(8)	6(7)	A		2	
SMARTQVT	RUBYTL	6(8)	5(9)	A		3	
SMARTQVT	UMT	6(8)	5(9)	A		3	
KMTL	MOMENT	6(9)	6(8)	A		2	
KMTL	GREAT	6(9)	6(10)	B		2	
KMTL	ATL	6(9)	6(7)	A		2	
KMTL	RUBYTL	6(9)	5(9)	A		3	
KMTL	UMT	6(9)	5(9)	A		3	
MOMENT	GREAT	6(8)	6(10)	B		2	
MOMENT	ATL	6(8)	6(7)	A		2	
MOMENT	RUBYTL	6(8)	5(9)	A		3	
MOMENT	UMT	6(8)	5(9)	A		3	
GREAT	ATL	6(10)	6(7)	A		2	
GREAT	RUBYTL	6(10)	5(9)	A		3	
GREAT	UMT	6(10)	5(9)	A		3	
ATL	RUBYTL	6(7)	5(9)	A		3	
ATL	UMT	6(7)	5(9)	A		3	
RUBYTL	UMT	5(9)	5(9)	A		1	

- Colocar la puntuación obtenida de la comparación por pares en cada una de las dimensiones para obtener los pesos locales o para establecer las prioridades locales entre los LTM.

Los resultados obtenidos de la comparación por pares (ver columna intensidad en las tablas anteriores) son colocados en sus correspondientes matrices de comparación para la

determinación o establecimiento de las prioridades locales entre los LTM, la tabla 13 muestra las prioridades locales asignadas a cada LTM con relación a cada una de las dimensiones, debido a limitaciones de espacio solo se muestra la dimensión formalización de lenguaje, los detalles para las otras dimensiones pueden verse en [13].

**Tabla 13.** Prioridades locales. Matriz de comparación por pares que muestra las preferencias para las 10 LTM en función de la dimensión formalización del lenguaje

LTM	JMI	OptimalJ	KERMETA	SmartQVT	KMTL	MOMENT	GREAT	ATL	RubyTL	UMT	Prioridad
JMI	1	1/3	¼	2	1	2	1/2	2	3	3	0,089
OptimalJ	3	1	½	3	3	3	3	3	5	5	0,194
KERMETA	4	2	1	4	4	4	3	4	5	5	<b>0,260</b>
SmartQVT	½	1/3	¼	1	1/2	1	1/2	2	3	3	0,069
KMTL	1	1/3	¼	2	1	2	1/2	2	3	3	0,089
MOMENT	½	1/3	¼	1	1/2	1	1/2	2	3	3	0,069
GREAT	2	1/3	1/3	2	2	2	1	2	3	3	0,111
ATL	½	1/3	¼	½	1/2	1/2	1/2	1	3	3	0,057
RubyTL	1/3	1/5	1/5	1/3	1/3	1/3	1/3	1/3	1	1	0,030
UMT	1/3	1/5	1/5	1/3	1/3	1/3	1/3	1/3	1	1	0,030

**Tabla 14.** Tabla resumen. Puntuación relativa de los LTM para cada Dimensión

	JMI	OPTIMALJ	KERMETA	SMARTQVT	KMTL	MOMENT	GREAT	ATL	RUBYTL	UMT
Formalización	0,089	0,194	<b>0,260</b>	0,069	0,089	0,069	0,111	0,057	0,030	0,030
Organización	0,120	0,040	0,212	0,046	0,028	0,057	0,072	0,064	<b>0,346</b>	0,015
Escenarios	0,038	0,072	0,122	0,145	0,049	0,145	0,059	<b>0,274</b>	0,038	0,059
Espacio	0,020	0,065	0,051	0,121	0,051	0,213	0,064	<b>0,365</b>	0,025	0,025
Aspectos	0,076	0,113	<b>0,162</b>	0,076	0,054	0,113	0,113	<b>0,162</b>	0,054	0,076
Arquitectura	0,077	0,077	0,077	0,077	0,077	<b>0,217</b>	0,029	0,077	0,077	<b>0,217</b>
Herramienta	0,072	0,066	0,081	0,139	0,040	<b>0,264</b>	0,049	0,190	0,049	0,049

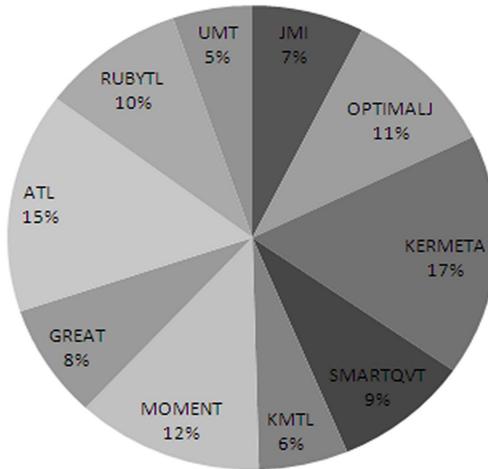
- Obtener la puntuación relativa de cada LTM respecto al objetivo general. Obtención de pesos globales para determinar cual alternativa es la más conveniente para la solución del problema planteado.

Para el cálculo de los valores o pesos globales, se consideran todos las dimensiones en conjunto a través de la sumatoria de la puntuación (prioridad relativa) obtenida localmente por el

LTM en cada una de las dimensiones (ver tabla 14) multiplicado por la puntuación (prioridad relativa) que aporta el peso específico según el aporte que tiene cada dimensión para satisfacer el objetivo (ver tabla 10); de allí se obtienen los valores globales para cada una de los LTM (ver tabla 15). Esto permite determinar cuál LTM es el más adecuado para la solución del problema planteado.

**Tabla 15.** Pesos globales

JMI	OPTIMALJ	KERMETA	SMARTQVT	KMTL	MOMENT	GREAT	ATL	RUBYTL	UMT
0,074	0,105	<b>0,168</b>	0,090	0,059	<b>0,124</b>	0,079	<b>0,153</b>	0,095	0,053



**Figura 2.** Grafico de jerarquías de los LTM

La figura 2 muestra un gráfico de torta que indica el tamaño de la tajada que tiene cada uno de los LTM evaluados. Es de destacar que los LTM Kermeta y ATL poseen los valores más altos, 17% y 15% respectivamente. Esto es consistente con los resultados obtenidos por el análisis de características, que indican que ambos LTM implementan una amplia gama de las características evaluadas.

#### 2.1.6.2 Análisis de resultados por dimensión

Se observa que los LTM MOMENT y SmartQVT satisfacen en gran medida la dimensión “escenarios de uso” con una distribución homogénea de atributos satisfechos, lo cual indica su versatilidad, seguido muy de cerca por ATL. Puede observarse que todos los LTM poseen todas las características obligatorias, sin embargo se evidencia que en algunos LTM estas características están mejor implementadas.

Por otra parte, el LTM RubyTL es el ganador para la dimensión “organización” destacándose en la diversidad de métodos para la definición de elementos destino, indicando su gran flexibilidad, seguida de cerca por Kermeta.

Así mismo, el LTM Kermeta es el indiscutible ganador para la dimensión “formalización” destacándose en su notación semántica formal, seguida de cerca por el LTM OptimalJ.

En la dimensión “espacio tecnológico” que determina cuáles lenguajes se aproximan más al estándar MDA y MOF 2.0/QVT, los que se destacan son los LTM ATL, MOMENT y SmartQVT.

En la dimensión “arquitectura” claramente se evidencia que el tipo de arquitectura utilizada en la mayoría de los LTM es la combinación interpretador/compilador.

Es notable indicar que en todos los lenguajes evaluados se satisfacen todas las características que representan la

dimensión “*aspectos no funcionales*”, sin embargo, los LTM KERMETA y ATL son los que han implementado mejor estas características con respecto a los otros lenguajes, debido a que ambos lenguajes poseen la mayor cantidad de atributos satisfechos.

En la dimensión “*herramienta de soporte*” trata sobre los entornos de desarrollo orientados al DSDM, donde el LTM ATL indiscutiblemente es el que posee una herramienta de apoyo que implementa todas las características en esta dimensión.

### III. CONCLUSIONES

Se ha definido un marco de evaluación de los LTM a través de un conjunto de dimensiones que representan características necesarias y sugeridas vinculadas con los requisitos que expresaron los autores según el análisis de sus trabajos en el área del DSDM. La evaluación realizada en este trabajo sigue un método de evaluación basado en la metodología DESMET, y particularmente sigue el enfoque de análisis de características por selección. Cabe destacar que el método definido fue combinado con el método AHP, el cual permitió que a partir de los resultados obtenidos al aplicar DESMET, establecer una jerarquía para precisar la escogencia de una de ellas. La jerarquía considera todas las dimensiones identificadas. Por otra parte los pesos son sujetivos: otro evaluador puede darle otros pesos variando así la jerarquía en función del interés de la dimensión que considere prioritaria, por lo cual los resultados podrían variar. Aunque existen varios trabajos relacionados dentro del enfoque DSDM que analizan las características que deben tener los LTM, ninguno contempla en conjunto todas las dimensiones analizadas en este trabajo. El hecho que un LTM tenga más atributos satisfechos en una cierta característica con respecto a una dimensión en comparación con otros LTM, indica que el LTM tiene una mejor implementación para esa característica.

Entre las características comunes encontradas: considerando la dimensión formalización están la forma de representación de las transformaciones, donde la forma gráfica ha probado ser la más popular; la variedad de construcciones sintácticas que posee el lenguaje; el estilo de transformación utilizado, que puede ser imperativo, declarativo o híbrido; el ámbito del lenguaje, donde un DSL proporciona notaciones adaptadas a un dominio de aplicación particular; en la dimensión organización se encuentran la modularidad, que permite combinar, agrupar y componer transformaciones para generar otras, útiles para aumentar la legibilidad y mantenibilidad; los mecanismos de integración que denotan un medio para el ensamblaje de un conjunto de reglas para el logro de nuevas funcionalidades; la correspondencia de elementos origen-destino al describir las reglas transformación; en la dimensión escenarios de uso tenemos la sincronización para verificar la coherencia entre modelos, refinamiento o actualización incremental, ingeniería inversa, generación de vistas particulares del sistema, aplicación de patrones de diseño, refactorización, entre otros; en la dimensión herramienta tenemos el tipo de soporte que ofrece

la herramienta en apoyo al lenguaje para la especificación de las transformaciones, como la trazabilidad que registra las correspondencias entre instancias de los modelos origen y destino, que permite el depurado; por último para la dimensión arquitectura se tiene la relación costos/beneficios influenciados por el tipo arquitectura subyacente seleccionada para ejecutar la transformación.

Las dimensiones permiten al desarrollador visualizar las fortalezas y debilidades en cada LTM, facilitando la toma de decisión en la elección del LTM acorde a las necesidades del desarrollador. Los resultados obtenidos son de ayuda para las personas que se inicien en el DSDM y en la eventual escogencia de un LTM.

### REFERENCIAS

- [1] Agrawal, A., 2003. GReAT: A Metamodel Based Model Transformation Language, Institute for Software Integrated Systems (ISIS), Vanderbilt University.
- [2] Akehurst, D. H., Howells, W. G. McDonald-Maier, K. D., 2005. Kent Model Transformation Language, University of Kent at Canterbury, UK.
- [3] Bézivin, J., Dupé, G., Jouault, F., Pitette, G., Rougui, J. E., 2003. First experiments with the ATL model transformation language: Transforming XSLT into Xquery, Atlas Group, INRIA and IRIN University of Nantes.
- [4] Czarnecki, K., Helsen, S., 2003. Classification of model transformation approaches. In Proceedings of the 2nd OOPSLA Workshop on Generative Technique in the Context of the Model Driven Architecture.
- [5] Czarnecki, K., Helsen, S. 2006. Feature-based survey of model transformation approaches. IBM Systems Journal, Vol. 45, No 3.
- [6] Czarnecki K., 2004. Overview of Generative Software Development. In Proceedings of Unconventional Programming Paradigms (UPP) 2004, 15-17 September, Mont Saint-Michel, France, Revised Papers, en <http://swen.uwaterloo.ca/~kczarnec/gsdoverview.pdf>
- [7] Cuadrado, J. S., Molina, J. J. G., Tortosa, M. M., 2006. Rubytl: Un Lenguaje de Transformación de Modelos Extensible, XV Jornadas de Ingeniería del Software y Bases de Datos. Barcelona.
- [8] Dupe, G., Belaunde, M., 2007. SmartQVT, France Telecom, <http://smartqvt.elibel.tm.fr/>
- [9] García, J. M., Rodríguez, J., Menárguez, M., Ortín, M.J., Sánchez, J., 2004. Un estudio comparativo de dos herramientas MDA OptimalJ y ArcStyler, Departamento de Informática y Sistemas, Universidad de Murcia.
- [10] Gardner, T., Griffin, C., Koehler, J., Hauser, R., 2003. A review of OMG MOF 2.0 Query/Views/ Transformations Submissions and Recommendations towards the final Standard.
- [11] Gerber, A., Lawley, M., Raymond, K., Steel, J., Wood, A., 2002. Transformation: The missing link of MDA, Graph Transformation: First International Conference, ICGT, <http://archive.dsc.edu.au/AU/staff/kerry-raymond/missing-link.pdf>
- [12] Grønmo, R., Oldevik, J., 2005. An Empirical Study of the UML Model Transformation Tool (UMT), In 1st Int. Conf. Interoperability of Enterprise Software and Applications. [http://interopesa05.unige.ch/INTEROP/Proceedings/Industrial/IND2\\_Gronmo.pdf](http://interopesa05.unige.ch/INTEROP/Proceedings/Industrial/IND2_Gronmo.pdf)