

GENERACIÓN AUTOMÁTICA DE APLICACIONES SOFTWARE A PARTIR DEL ESTANDAR MDA BASÁNDOSE EN LA METODOLOGÍA DE SISTEMAS EXPERTOS E INTELIGENCIA ARTIFICIAL

AUTOMATIC GENERATION OF SOFTWARE APPLICATIONS FROM STANDARD MDA STANDARD BASED ON THE METHOD OF ARTIFICIAL INTELLIGENCE AND EXPERT SYSTEMS



AUTOR

IVÁN MAURICIO RUEDA CÁCERES
Ingeniero de Sistemas
Universidad Industrial de Santander
ivanruedac@gmail.com
COLOMBIA

INSTITUCIÓN

GOMEZ CAJIAO Y ASOCIADOS
GOMCA
Empresa de Servicios de Ingeniería
Carrera 19 B N° 82-46, Bogotá
sales@gomca.com
COLOMBIA

RECEPCIÓN: Noviembre 15 de 2010

ACEPTACIÓN: Febrero 11 de 2011

TEMÁTICA: Nuevas Herramientas y Tecnologías de Desarrollo Software

TIPO DE ARTÍCULO: Corto

RESUMEN ANALÍTICO

Son muchos los estudios que se han presentado a cerca de la generación automática de líneas de código, este artículo pretende presentar una solución a las limitaciones de una herramienta muy conocida llamada MDA, haciendo uso los avances tecnológicos de la inteligencia artificial y los sistemas expertos. Abarca los principios del marco de trabajo de MDA, transformando los modelos usados y añadiendo características a estos que permitirán hacer más eficiente esta metodología de trabajo. El modelo propuesto abarca las fases del ciclo de vida software siguiendo las reglas del negocio que hacen parte esencial un proyecto real de software. Es con las reglas del negocio que se empieza a dar la transformación del estándar MDA y se pretende dar un aporte que contribuya a automatizar las reglas del negocio de forma tal que sirva para la definición de las aplicaciones en todo el ciclo de vida que la genera.

PALABRAS CLAVES: Automatización de aplicaciones software, UML, MDA: CIM, PSM, reglas de negocio, sistema experto, inteligencia artificial, ingeniería del software.

ANALYTICAL SUMMARY

Many studies are presented about automatic generation of code lines, this article want to present a solution for limitations of a tool called MDA, using from Artificial intelligence technological advances and expert systems. covering the principle of MDA work frame, transforming used models and adding characteristics to this that allow to make more efficient this work methodology. the proposed model covers the phases cycle life software, following the business rules that make essential part in a real software project. With the Business rules can start to transform the standard MDA aiming to give a contribution to automate the business rules that works to define applications in all the life's cycle that generate it.

KEY WORDS: Automation applications software, UML, MDA: CIM PSM, business rules, expert system, artificial intelligence, software engineering.

INTRODUCCION

Este artículo empieza abordando la generación automática de software desde la perspectiva de las organizaciones que hacen uso de las reglas del negocio para alcanzar sus metas y trazar sus políticas y estrategias, como punto neurálgico a la hora de desarrollar aplicaciones puesto que no son integradas en forma tangible a los procesos del ciclo de vida de la aplicación software.

Luego de esto, se adentra en el modelado con UML ya que son los modelos de diseño utilizados para la generación de código. Para luego realizar la transformación al MDA, que consta de una propuesta que incluye agentes como base para mejorar este framework. A su incluye una forma de acoplar los principios de los sistemas expertos para aplicarlos en la fase anterior a la generación automática de software.

1. REGLAS DE NEGOCIO Y MDA

El factor deliberante al momento de construir un generador de código, es la automatización de la codificación de las reglas del negocio. Los avances hasta ahora sobre generación de código tratan solamente aquellas capas de una arquitectura que no pertenecen a la codificación de las reglas del negocio, por ejemplo:

La capa de acceso a datos (procedimientos de insertar, editar, eliminar y actualizar).

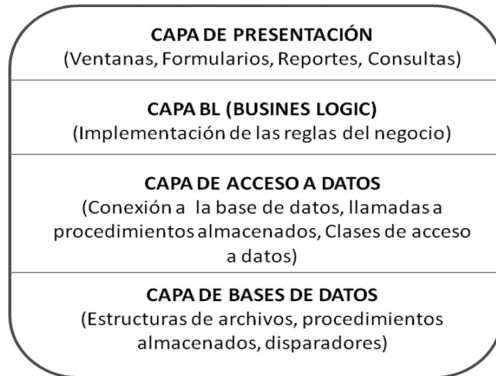
La capa de base de datos (Scripts de creación de la base de datos y sus objetos).

La capa de presentación (estándares previamente definidos).

Para la capa de las reglas del negocio (BL Business Logic),

las aplicaciones solamente generan la declaración de las clases con sus atributos y la declaración de los métodos. Es en esta capa donde se encuentra la complejidad de la codificación de software; y más aún cuando esta afecta el comportamiento de los eventos en formularios y ventanas.

FIGURA 1. Capas de la arquitectura de software



Las reglas del negocio se pueden representar a través de los modelos UML ya que en ellos se plasman el análisis y diseño de las aplicaciones a implementar. "UML es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables" [I].

Para plasmar las reglas del negocio en UML es necesario hablar del estándar MDA, el cual consiste en incluir modelos relacionados entre sí, establecidos en distintos niveles de abstracción, con la trazabilidad definida de un conjunto de uno modelo hacia otro que incluye:

Un CIM¹ que contenga todas las reglas del negocio definidas en el modelo del proceso de negocio.

Un PIM² que defina el modelo conceptual completo con todas sus relaciones.

Uno o varios PSM³ para generar componentes de ejecución y pruebas del sistema.

2. GENERACIÓN AUTOMÁTICA DE APLICACIONES SOFTWARE

La generación automática de código es una idea sobre la cual se ha venido investigando tiempo atrás, y que estándares como el MDA (Model-Driven Architecture) y el MDD (Model-Driven Development) han aportado

progresos que contribuyen mejoramiento de esta idea.

"MDA (Arquitectura Dirigida por Modelos), un concepto promovido por la OMG (Object Management Group), que propone basar el desarrollo de software en modelos especificados utilizando UML, para que, a partir de esos modelos, se realicen transformaciones que generen código u otro modelo, con características de una tecnología particular (o con menor nivel de abstracción). MDA define un framework para procesar y relacionar modelos. Suele escucharse que MDA es la evolución natural de UML, ya que tiende a incrementar la cantidad de código generado, a partir de especificaciones detalladas en UML", [II].

MDA es un framework para construir sistemas basándose en modelos UML, representa en diferentes capas de abstracción el sistema a implementar. A diferencia del Análisis y Diseño Orientado a Objetos (OOAD) el MDA adiciona una capa de abstracción más, que representa las reglas de negocio del sistema.

Entre las herramientas MDA más conocidas podemos encontrar:

- AndroMDA
- Enterprise Architect
- Rational Architect
- OptimalJ is a MDA tool for J2EE.
- MDA Transf
- ArcStyler is a MDA tool for J2EE and .NET.
- Entre otras

Las herramientas MDA nos proporcionan ahorro en la escritura y generación de líneas de código. El problema unánime de estas herramientas MDA es que no suministran la capacidad de transformar modelos de negocios (CIMs Modelo Independiente de la Computación) en aplicaciones listas para ejecutar sin necesidad de intervención técnica (programación). Por lo tanto el objetivo de la presente pre-propuesta doctoral es proveer una transformación de modelos que brinde las facilidades para automatizar cualquier negocio y problemática a desarrollar.

3. SISTEMAS EXPERTOS, AGENTES Y MDA

En MDA, los procesos del negocio son considerados la parte fundamental de los modelos CIM, para apoyar estos modelos se utilizara el estándar BPM, para la

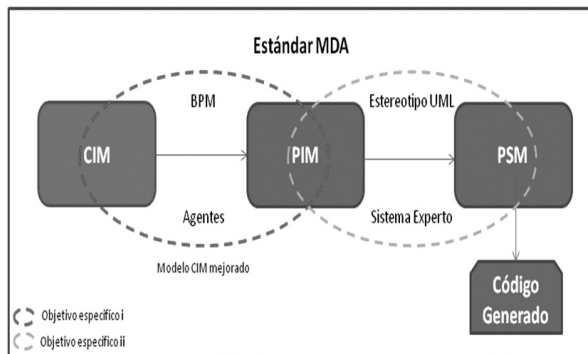
¹ Computation Independent Model (Modelo independiente de computo)

² Plataforma Independent Model (Modelo independiente de la plataforma)

³ Plataforma Specific Model (Modelo específico a la plataforma)

correcta definición de los procesos del negocio y para la unión entre los procesos del negocio y el software.

FIGURA 2. Proceso del Estándar MDA



3.1 TRANSFORMACIÓN DEL FRAMEWORK MDA

Para hablar de generación de código fuente, nos debemos remitir a una de las disciplinas más jóvenes "la inteligencia artificial" que ha evolucionado con el paradigma de los Agentes⁴ ("Los agentes constituyen el próximo avance más significativo en el desarrollo de sistemas y pueden ser considerados como la nueva revolución en el software"⁵), los avances hasta el momento suponen que la orientación de los futuros sistemas inteligentes es variado resolviendo problemas tanto en el ámbito académico como en el industrial.

El agente consta de un programa agente⁶ y una arquitectura⁷; una estructura de un agente podría ser la siguiente, donde se da aceptación a las reglas del negocio y se realiza la transformación de dichas reglas.

Aplicando la definición de "agente", para la transformación del modelo CIM debería estar compuesto de:

Programa agente: almacenador de reglas del negocio con modelado BPM, donde percibe y almacena una representación de las mismas.

Arquitectura: Transformador de las reglas del negocio (modelos CIM a PIM).

Otro concepto fundamental en la generación de automática de código son los sistemas expertos, ya que uno de sus finalidades es crear software a partir de otro software.

Los sistemas expertos recopilan y almacenan la sapiencia de un individuo en un campo específico. Con esta información los sistemas expertos pueden resolver problemas, tomar decisiones e incluso refinar su comportamiento y a la vez aumentar su conocimiento en dicho campo. Asimilando la idea de sistemas expertos

se podría decir que este encapsule la pericia de un programador experto y pueda generar una aplicación software, el punto clave es si el sistema experto puede en teoría cumplir con este objetivo. Si es posible aunque la idea suene un poco lejana, además imagínese el tiempo que se ahorraría omitiendo la fase de implementación del software ya que se generaría automáticamente, y lo fuerte que se volvería la fase de diseño quitando la mala práctica de las empresas de desarrollo software que no invierten el tiempo y los recursos necesarios a tan importante fase del ciclo de ingeniería del software.

Es importante destacar que el sistema experto tomará los conocimientos ya no de un individuo sino de un diseño estructurado en un estereotipo UML (ver figura 2) teniendo en cuenta la arquitectura MDA, de la cual nace la construcción de modelos de software, desde diferentes puntos de vista del negocio, como artefactos fundamentales en la implementación de software. Además la arquitectura es crucial en la transformación de modelos para automatizar la derivación a código ejecutable.

MDA se basa en desligar la especificación de la funcionalidad del negocio de la plataforma tecnológica determinada, fortaleciendo de esta manera la portabilidad, interoperabilidad y reutilización.

Por lo tanto es viable la construcción de un sistema experto que contenga el conocimiento y la experiencia de un habilidoso programador, basándose en las reglas del negocio que se han especificado en el modelo CIM mejorado (modelo con BPM y agentes) diseñados para la aplicación software teniendo en cuenta el estándar MDA.

Al llevar a cabo con éxito el planteamiento de esta propuesta doctoral, se lograría una revolución en la ingeniería del software y por ende en la industria colombiana de software aumentando significativamente la calidad de los productos al eliminar el factor humano en la fase de codificación, ya que se reduciría los tiempos de entrega y los costos de fabricación, además de crear un ámbito estructurado de la fase de diseño del ciclo de vida del producto la cual en muchos casos es débil en las empresas colombianas. Fortaleciendo el posicionamiento de la industria colombiana de software en los mercados internacionales y haciéndola altamente competitiva.

⁴ Entidad que percibe y actúa sobre un entorno [Russell1996]

⁵ Por el Dr. Nicholas Jennings

⁶ Función que implementa la transformación (mapping) de secuencias de percepciones en acciones

⁷ Ordenador que se ocupa de que las percepciones lleguen al programa y las acciones lleguen a los efectores (la estructura encargada de ejecutar la acción)

4. METODOLOGÍA

Se propone la metodología compuesta por las siguientes fases:

- A.** Caracterización de la generación automática de software: Revisión de los principales sistemas, procesos y técnicas utilizadas para la generación automática de software.
- B.** Revisión del marco teórico y estado del arte: Revisión de la literatura sobre sistemas expertos, inteligencia artificial, modelos BPM y estándar MDA y las aplicaciones y casos de éxito en la generación automática de software.
- C.** Especificación de Requerimientos: Análisis y elicitación de los requerimientos para el sistema experto generador de aplicaciones.
- D.** Realizar el modelo CIM mejorado propuesto: Presentación y revisión del modelo CIM mejorado, pruebas de consistencia y revisión por expertos.
- E.** Realizar un modelo del sistema experto: Presentación y revisión del modelo del sistema experto generador de aplicaciones, pruebas de consistencia y revisión por expertos.
- F.** Establecer la estructura del programa agente como modelador de las reglas del negocio, pruebas de consistencia y revisión por expertos.
- G.** Cierre y Divulgación: Elaboración de conclusiones, revisión de los documentos de divulgación.

5. PRINCIPALES ACTIVIDADES A DESARROLLAR

Trabajo de investigación:

- a)** Revisión de las tendencias, sistemas y herramientas utilizados para el ejercicio de la generación automática de aplicaciones.
- b)** Análisis del estado del arte de la generación automática de aplicaciones software, análisis de los principales artículos y los trabajos de investigación desarrollados.
- c)** Consulta a expertos del tema.
- d)** Estudio de las relaciones existentes entre las sistemas expertos, inteligencia artificial y MDA y modelos BPM.
- e)** Modelamiento del CIM mejorado.
- f)** Modelamiento del sistema experto.
- g)** Pruebas de consistencia de las transformaciones propuestas.

6. CONCLUSIONES

Fortalecer la línea de investigación en Ingeniería del software mediante el desarrollo de nuevos modelos que aplicados a las herramientas existentes contribuya a mejorarlas y nos brinden nuevas posibilidades de actuación en las disciplinas de la gestión de la información y las TICS.

Desarrollar experticia en inteligencia artificial, sistemas expertos, modelos BPM, estándar MDA y UML para compartirlos con la academia y los miembros de los grupos que desarrollan trabajos en la línea de investigación en ingeniería del software.

Fomentar lazos de cooperación entre grupos de investigación y la industria colombiana de software con trabajos de investigación aplicada, orientada al mejoramiento de la calidad de los productos y procesos de Ingeniería del software.

7. MÉTODOS (RECOGIDA DE INFORMACIÓN, MATERIALES, ETCÉTERA)

Antes que nada se debe definir claramente el tipo de investigación que se va a llevar a cabo para ello hay que recolectar de forma correcta los datos a utilizar teniendo en cuenta el planteamiento del problema. Teniendo en cuenta lo anterior, para este caso se debe realizar lo siguiente:

- a) Seleccionar el diseño de investigación.
- b) Recolectar información que sea relevante para el modelo a desarrollar.
- c) Establecer parámetros de cualificación.
- d) Desarrollar la metodología a seguir para desarrollar el modelo.

9. REFERENCIAS

- [I] UML y Patrones, Craig Larman, Prentice Hall, 2 edición 2003
- [II] Revista Software Gurú conocimiento en práctica, N°. 19 Febrero-Abril 2008 pag. 36
- [III] Artículo "Artículo "BOM-Lazy: gestión de la variabilidad en el desarrollo de Sistemas Expertos mediante técnicas de MDA", María Gómez, Abel Gómez, M^a Eugenia Cabello, Isidro Ramos, Universidad Pontificia de Valencia y Universidad de Colima.
- [IV] Object Management Group <http://www.omg.org/>

- [V] Generación automática de restricciones de integridad <http://jordicabot.com/papers/DSDM05.pdf>
- [VI] "Executable UML, A Foundation for Model-Driven Architecture", S. J. Mellor, M. J. Balcer
- [VII] "MDA Explained, The Model Driven Architecture: Practice and Promise", A. Kleppe, J. Warmer, W. Blast
- [VIII] Business Process Modeling <http://www.bpmodeling.com/>