
**Departamento de Tecnologías y
Sistemas de Información**

**Model Driven Engineering
Aplicado a
Business Process Management**

Jose Manuel Pérez, Francisco Ruiz, Mario Piattini

Informe Técnico UCLM-TSI-002
Marzo 2007



Universidad de
Castilla-La Mancha

ÍNDICE

ÍNDICE	i
ÍNDICE DE FIGURAS	iii
ÍNDICE DE TABLAS	iv
RESUMEN	1
1. INTRODUCCIÓN	3
1.1. ¿Qué es el paradigma MDE?	3
1.1.1. Un ejemplo de MDE: MDA	4
1.1.1.1. Tipos de modelos MDA	5
1.1.1.2. Transformaciones de modelos	7
1.2. ¿Qué es el BPM?	8
1.2.1. BPMS. Sistemas de Gestión de Procesos de Negocio	9
2. ESTÁNDARES DE PROCESOS DE NEGOCIO	11
2.1. BPMN (Business Process Modelling Notation)	11
2.2. BPML (Business Process Modelling Language)	12
2.3. BPDM (Business Process Definition Metamodel)	12
2.4. SBVR (Semantics of Business Vocabulary and Business Rules)	12
2.5. BPEL4WS (Business Process Execution Language for Web Services)	13
3. MDE APLICADO A BPM	13
3.1. Propuestas de aplicación de MDA a BPM	14
3.1.1. MDA como nexo de unión de los procesos de negocio entre varias organizaciones.	14
3.1.2. MDA como soporte al ciclo de vida en BPM	16
3.2. BPDM. El estándar de MDA para BPM	18
3.3. Estándares BPM para los distintos tipos de modelos MDA	19
3.4. Limitaciones	21
REFERENCIAS	22
ACRÓNIMOS	23

ÍNDICE DE FIGURAS

FIGURA 1. Dibujo representativo de las áreas y tecnologías que abarca MDA [3].....	5
FIGURA 2. Modelos CIM, PIM y PSM y su relación de transformación	6
FIGURA 3. Pirámide de metamodelos y modelos MOF [1].....	7
FIGURA 4. Esquema de operaciones dentro del BPM y sus relaciones [5].	8
FIGURA 5. Ejemplo de diagrama de procesos de negocio usando BPMN [9].....	12
FIGURA 6. Fragmento de código BPEL4WS para definir un proceso, usando notación XML.	14
FIGURA 7. Representación de los procesos de negocio colaborativos usando la arquitectura MDA [12].	15
FIGURA 8. Representación de los procesos de negocio colaborativos usando la arquitectura MDA, una vez elegidos los estándares adecuados [12].	16
FIGURA 9. Proceso de modificación de un proceso de negocio usando MDA.	17
FIGURA 10. Visión del metamodelo BPDM respecto al resto de metamodelos [13]... ..	19
FIGURA 11. Utilización de MDA en el proceso de desarrollo BPM.	20

ÍNDICE DE TABLAS

Tabla 1. Clasificación de los estándares de BPM según varios parámetros.....	20
--	----

RESUMEN

En el contexto de las organizaciones, la gestión de procesos de negocio (BPM) ha surgido para encauzar los continuos cambios que dichas organizaciones sufren en su cadena de valor y por lo tanto en sus procesos de negocio. La gestión de procesos de negocio ayuda a los analistas de negocio a manejar todos los aspectos relacionados con los procesos de negocio, pero el hueco entre estos analistas y la gente que construye las aplicaciones es demasiado grande. La ingeniería dirigida por modelos (MDE) aparece como una gran ayuda para transferir los cambios en los procesos de negocio a los sistemas que implementan dichos procesos. Así, usando alguna propuesta MDE, como MDA, se podría mejorar la interacción entre la gente de negocios y los ingenieros del software. En ese marco se encuadra el siguiente informe técnico, donde se recogen las diferentes propuestas y opiniones que distintos autores han escrito sobre el tema, así como una serie de estándares que ayudan al diseño de los procesos de negocio y que encajan dentro de la filosofía MDE.

1. INTRODUCCIÓN

En este apartado se hace una breve introducción a los dos conceptos cuya intersección aborda este informe técnico. Por un lado se introducirá el paradigma “Model Driven Engineering” (en adelante MDE), haciendo hincapié en “Model Driven Architecture” (en adelante MDA), creado por OMG como ejemplo de manera de llevar a cabo MDE. Por el otro, se pondrá en situación el “Business Process Management” (en adelante BPM) como filosofía de gestión y mejora de los procesos de negocio de las empresas.

1.1. ¿Qué es el paradigma MDE?

Durante las últimas dos décadas, los avances en lenguajes y plataformas han aumentado el nivel de abstracción disponible en la tarea de desarrollo de software. Además debido a la madurez de los lenguajes de tercera generación, los desarrolladores de software están mejor equipados para afrontar y resolver los distintos problemas que se les pueden plantear.

A pesar de todos estos avances, aún quedan problemas importantes que resolver. En el centro de éstos se encuentra el crecimiento de la complejidad de las plataformas, las cuales contienen miles de clases y métodos con dependencias muy complicadas que deben ser conocidas por el desarrollador. El problema se acrecienta cuando estas mismas plataformas crecen rápidamente y además aparecen otras nuevas, con el consiguiente esfuerzo de migración de unas a otras.

En este último caso, cuando la evolución tecnológica de las plataformas o de los sistemas se produce, es importante conservar el mismo modelo conceptual del negocio, es decir, que la lógica del dominio del problema debería ser la misma, sea cual sea la plataforma o lenguaje que implementa dicha lógica.

Para manejar el problema del crecimiento de la complejidad de los sistemas, la orientación a objetos no parece ser suficiente. Los lenguajes orientados a objetos han ido perdiendo la simplicidad con la que fueron ideados, la encapsulación no es un recurso tan útil como en principio parecía y, sobre todo, la reutilización de los objetos como componentes no ha tenido demasiado éxito en la industria del software [1].

Parece que las propuestas centradas en código no dan respuesta a las demandas de los sistemas actuales. Esta es la razón por la que ha aparecido una nueva propuesta centrada en modelos.

A esta propuesta se le llama Ingeniería Dirigida por Modelos (MDE). Este paradigma combina los siguientes conceptos [2]:

- *Lenguajes de dominio específico*. Formalizan la estructura de la aplicación, el comportamiento y los requisitos dentro de un dominio particular. Estos lenguajes (DSL) son descritos usando metamodelos, los cuales definen relaciones entre elementos dentro de un dominio.

- *Motores de transformación y generadores*. Analizan ciertos aspectos de los modelos después crean varios tipos de artefactos, tal como código fuente, entradas de simulación, descripciones de uso XML, o representaciones alternativas de dicho modelo.

Las herramientas MDE usan los conceptos anteriores y hacen más fácil para los ingenieros del software el soporte a la evolución del software, tanto en su lógica como en su tecnología.

Mediante los DSL's se consiguen notaciones de modelado distintas para cada tipo de sistema, las cuales están definidas formalmente por su metamodelo. De esta manera, el ingeniero del software tiene herramientas específicas para cada tipo de sistema, lo cual le permite modelarlos de una manera más detallada y de acuerdo al dominio al que pertenecen.

Mediante los motores de transformación se facilita la evolución de modelos, transformando de unos modelos a otros, según la reglas de transformación entre metamodelos.

En el paradigma MDE cualquier concepto debe ser modelado. De esta manera, cualquier cambio o nueva propiedad del sistema debe ser mostrado en su modelo correspondiente. Con este paradigma, la parte de escritura de código es una parte más del proceso de construcción de sistemas (quizás la menos importante), la cual se sugiere que se realice automáticamente.

1.1.1. Un ejemplo de MDE: MDA

El consorcio OMG (Object Management Group) ha desarrollado la propuesta “Arquitectura Dirigida por Modelos” o MDA como ejemplo de implementación de MDE.

MDA (ver figura 2) nace con la idea establecida de separar la especificación de la lógica operacional de un sistema, de los detalles que definen cómo el sistema usa las capacidades de la plataforma tecnológica donde es implementado.

Teniendo en cuenta lo anterior, los objetivos de MDA son la portabilidad, la interoperabilidad y la reusabilidad a través de la separación arquitectural.

El concepto de *independencia de plataforma* aparece frecuentemente en MDA. Es la cualidad que tienen los modelos de ser independientes de las características de cualquier tipo de plataforma tecnológica.

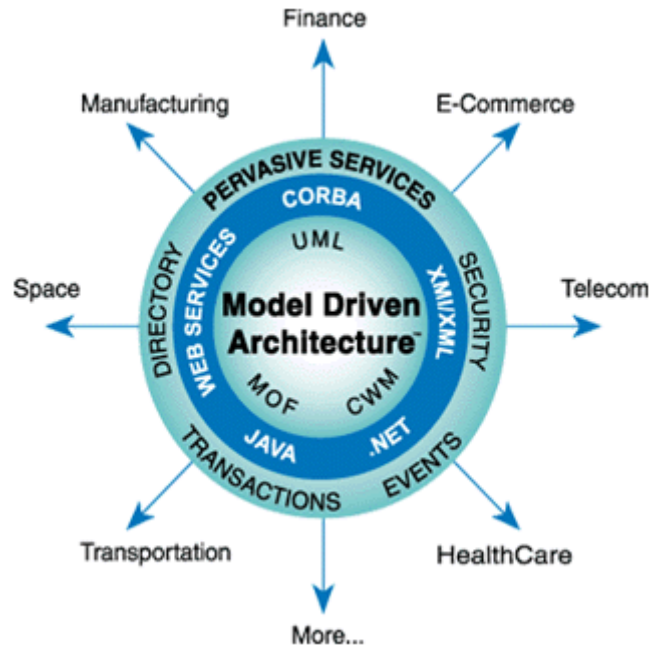


FIGURA 1. Dibujo representativo de las áreas y tecnologías que abarca MDA [3].

Mediante la aplicación de este paradigma se cubre completamente el ciclo de vida de un sistema software, desde la captura de requisitos hasta el mantenimiento del mismo, pasando por la generación del código fuente. Para ello define tres tipos de modelos que se explican a continuación.

1.1.1.1. Tipos de modelos MDA

Los tipos de modelos que MDA propone son los siguientes (figura 2):

- **Modelo Independiente de la Computación (CIM):** Un CIM no muestra detalles de la estructura del sistema. A veces es llamado modelo de dominio o modelo de negocio. En el CIM se modelan los requisitos que deberá satisfacer el sistema, describiendo la situación en la cual el sistema será usado. Es muy útil tanto para ayudar a comprender el problema como para ejercer de fuente de vocabulario compartido para el uso en otros modelos. Según MDA, la especificación de requisitos de un sistema CIM debería ser transformable en un PIM y posteriormente en un PSM y viceversa. El CIM juega un papel importante como puente entre los que son unos expertos en el dominio del problema y sus requisitos y aquellos que son expertos en el diseño y construcción de artefactos software.
- **Modelo Independiente de la Plataforma (PIM):** Un modelo PIM muestra el grado de independencia de plataforma necesario para poder ser usado en diferentes plataformas tecnológicas de un tipo similar. Este modelo debe tener tal nivel de abstracción que no cambie, sea cual sea la plataforma elegida para su implementación. Con este modelo se representa la lógica del sistema y sus interacciones con el mundo exterior, sin entrar en detalle de que tipo de tecnología implementará cada parte y cómo se adapta a una plataforma específica.

- **Modelo Específico de Plataforma (PSM):** El modelo PSM es una vista del sistema para una plataforma específica. Éste combina la especificación del sistema hecha en el PIM, con los detalles que especifican la manera en que dicho sistema usa una plataforma particular.

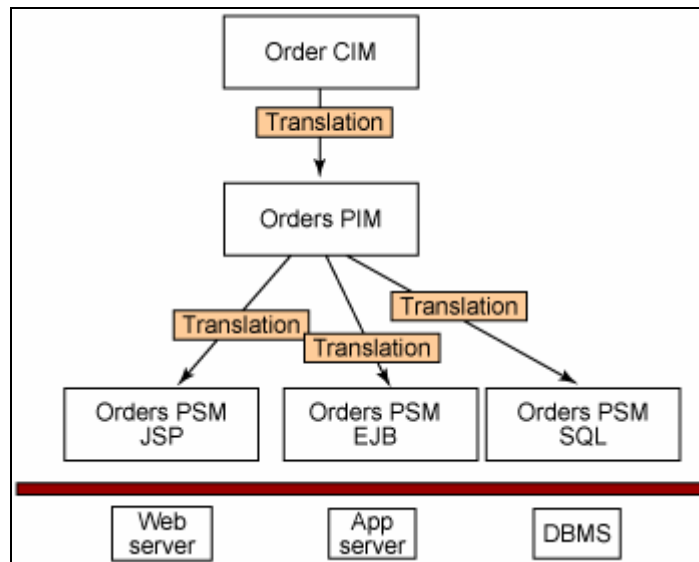


FIGURA 2. Modelos CIM, PIM y PSM y su relación de transformación.

Aunque UML es el lenguaje de modelado central de MDA, no todos los modelos tienen por qué estar especificados en dicho lenguaje. Para ello entra en juego el concepto de *metamodelo*. Un *metamodelo* es un modelo para definir modelos. UML es un metamodelo que especifica cómo crear modelos UML. Es decir, un modelo UML es una instancia del metamodelo UML.

Como se ha dicho, MDA no implica el uso de UML, pero en su lugar la tecnología crucial es MOF y la definición de metamodelos que sean instancias del meta-metamodelo MOF [1].

Cada uno de estos metamodelos define un lenguaje de modelado de dominio específico, que presenta una solución al modelado de distintos tipos de sistemas software. Por ejemplo, existe el metamodelo UML para modelar la arquitectura de sistemas discretos orientados a objetos, o el metamodelo SPEM para modelar procesos software, etc. Todos estos metamodelos son a su vez instancias del meta-metamodelo MOF.

Para entender bien este concepto en la figura 3 se muestra la pirámide de niveles de modelado de la arquitectura conceptual de MOF.

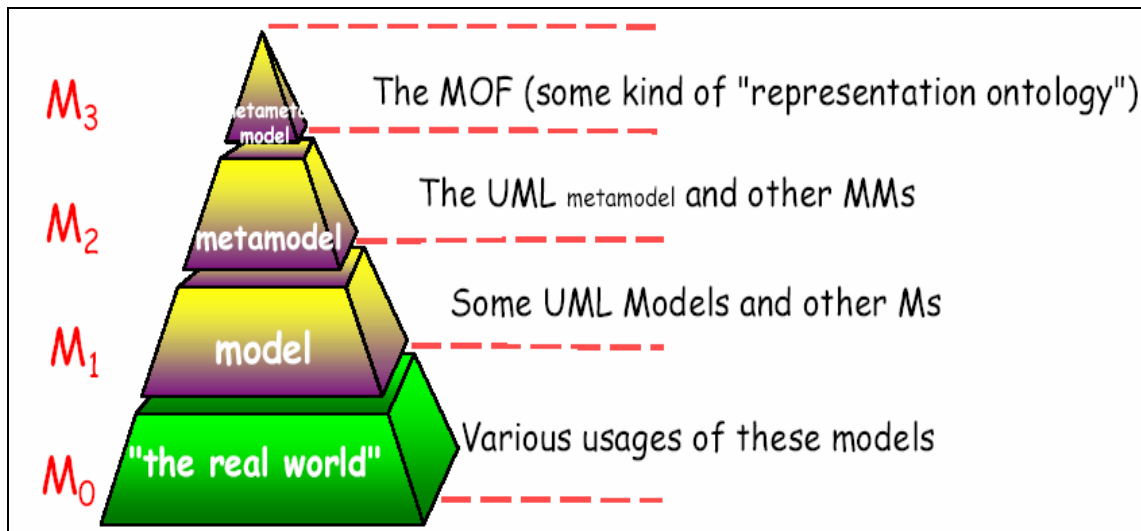


FIGURA 3. Pirámide de metamodelos y modelos MOF [1].

1.1.1.2. Transformaciones de modelos

La transformación de modelos es el proceso de convertir un modelo en otro modelo del mismo sistema.

En el paradigma MDA, la transformación de modelos puede ser horizontal o vertical. Una transformación horizontal consiste en pasar un modelo de un nivel de abstracción M_x a otro modelo del mismo nivel M_x , pero ambos basados en un modelo del nivel inmediato superior M_{x-1} diferentes. Una transformación vertical ocurre cuando los modelos pertenecen a dos niveles inmediatos diferentes, M_x y M_{x-1} . Por ejemplo, se puede usar la transformación vertical de modelos para pasar de un modelo PIM a un modelo PSM, o de un modelo CIM a un modelo PIM; o una transformación horizontal para pasar de un PIM1 a otro PIM2 basado en un metamodelo diferente que el primero.

Las distintas etapas del ciclo de vida del software pueden ser representadas en función de los distintos tipos de modelos que MDA propone y de la transformación de dichos modelos con la transición de una etapa a otra.

Para el caso en que se aplique el paradigma MDA a un sistema heredado (legado o antiguo), se puede aplicar la reingeniería para soportar su evolución, para lo que es necesario la transformación inversa de modelos (de PSM a PIM), lo cual también es sugerido por MDA.

El lenguaje estándar que OMG propone para la definición de transformaciones de modelos es el lenguaje QVT, que a su vez se basa en el lenguaje de restricciones OCL (*Object Constraint Language*). Con QVT se pueden definir transformaciones genéricas entre metamodelos, así cualquier instancia del metamodelo fuente puede ser transformado en una instancia del metamodelo destino.

1.2. ¿Qué es el BPM?

Un proceso de negocio es una colección de actividades que, tomando una o varias clases de entradas, crea una salida que tiene valor para un cliente [4].

La gestión de procesos de negocio o BPM se define como la habilidad de descubrir, diseñar, desplegar, ejecutar, interactuar, operar, optimizar y analizar completamente procesos y hacerlo al nivel de diseño de negocio, no de implementación técnica [5].

BPM es la disciplina para modelar, automatizar, gestionar y optimizar procesos de negocio para incrementar la rentabilidad.

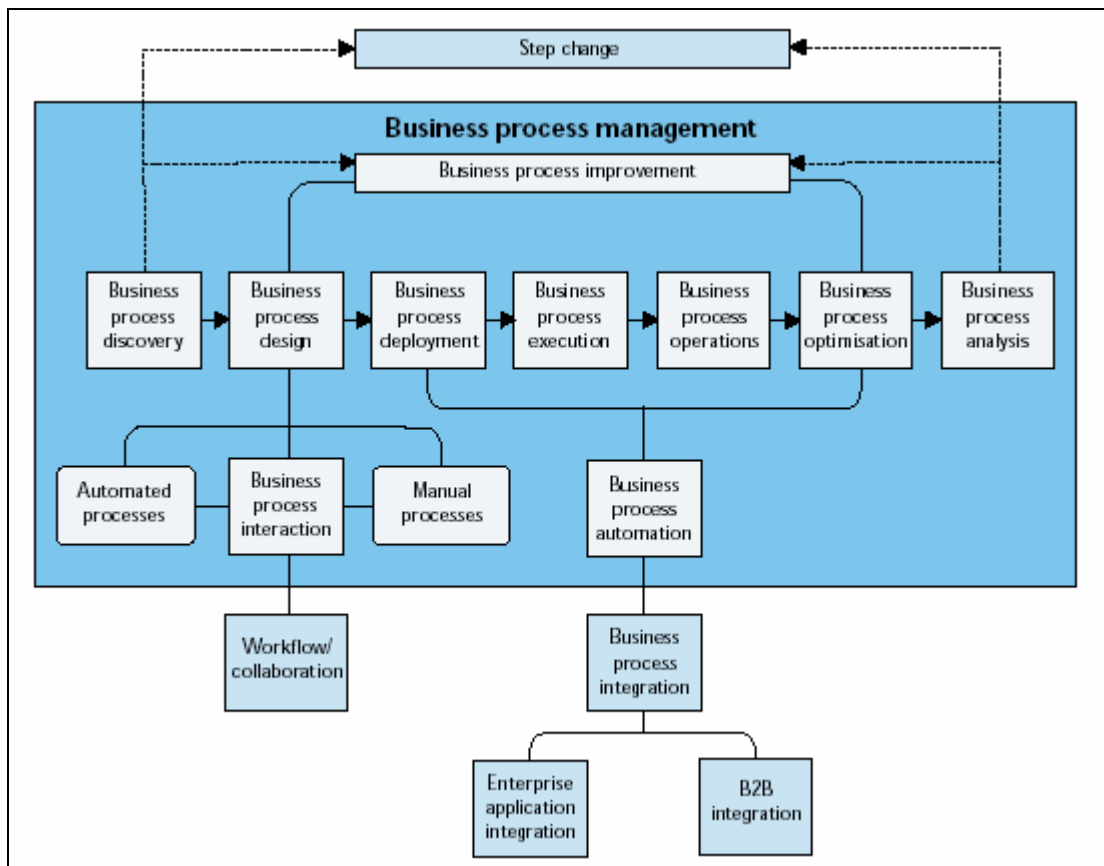


FIGURA 4. Ciclo de vida de los procesos de negocio según el paradigma BPM [5].

Los beneficios de adoptar BPM en una organización son los siguientes:

- **Mejora de la velocidad de realización de los procesos de negocio.** BPM puede reducir los tiempos reduciendo las demoras y las duraciones de las tareas mediante la automatización de ciertos pasos, permitiendo que varias etapas se den en paralelo e imponiendo límites de tiempo en la terminación de las tareas.
- **Incremento de la satisfacción del cliente.** Acelerando los procesos y asegurando que nada falla, tanto los clientes internos como los externos obtienen la información y las respuestas que necesitan más rápida y fácilmente.

- **Responsabilidad e integridad.** BPM asegura que todas las reglas de negocio requeridas son satisfechas y todos los pasos completados.
- **Optimización y eliminación de tareas innecesarias.** Simplemente modelando los procesos, las organizaciones pueden frecuentemente encontrar oportunidades y eliminar trabajo innecesario. Además usando un BPMS (explicado más adelante), se pueden proporcionar medidas de los procesos que se están gestionando facilitando el seguimiento y control de los mismos, así como su mejora y optimización.
- **Inclusión de clientes y socios de mercado en los procesos de negocio.** BPM permite a clientes y socios participar activamente en los procesos de negocio de una organización. Esto hace que las posibilidades de colaboración aumenten, haciendo que la distancia física no sea un impedimento.
- **Agilidad organizacional.** BPM proporciona un excelente medio para conseguir agilidad organizacional. Cuando un proceso cambia (algo muy común en las organizaciones), es relativamente fácil cambiar las reglas, los roles y las relaciones que definen ese proceso.

BPM no está directamente relacionado con el desarrollo de aplicaciones software. Su principal interés es gestionar los procesos de negocio, aunque esto requiera ayuda de la informática. Los modelos formales de procesos de negocio son entendibles por una máquina, y además las herramientas que existen alrededor del BPM pueden presentar estos modelos de manera que la gente de negocio pueda crearlos, leerlos y modificarlos.

BPM representa la “tercera ola” en la Ingeniería de Procesos de Negocio. La primera ola fueron en gran parte procesos que reorganizaban las actividades de las personas. La segunda ola se centró en la reingeniería de procesos de negocio y en el uso de las aplicaciones ERP, pero la reingeniería de procesos no era más fácil de gestionar y cambiar que lo que reemplazaba. La tercera ola se centra en los modelos formales de procesos de negocio y la capacidad de modificarlos rápidamente y de combinar esos modelos para rápidamente alinear los procesos de negocio con las estrategias cambiantes de las organizaciones.

En las dos primeras olas, ya se usaba el modelado de procesos de negocio pero sólo para fomentar la comprensión humana y no para dirigir la gestión de los procesos de negocio, como actualmente se pretende [6].

1.2.1. BPMS. Sistemas de Gestión de Procesos de Negocio

Un BPMS (*Business Process Management System*) es una nueva plataforma TI construida para gestionar procesos de negocio.

Existe una analogía entre un DBMS (Sistema Gestor de Bases de Datos) y un BPMS. Los datos son persistentes en un DBMS. Los procesos son persistentes en un BPMS. Un DBMS necesita ser fiable para gestionar los datos de negocio al igual que un BPMS necesita ser fiable para gestionar los procesos de negocio.

También existen diferencias como el hecho de que los procesos no son estáticos, sino que son activos, al contrario que los datos que son estáticos y no cambian hasta que son cambiados por un programa. Por lo tanto, un BPMS debe tener una máquina virtual de procesos que progresa o ejecuta los procesos de acuerdo a su diseño [7].

Con los BPMS se espera conseguir [5]:

- **Integración de sistemas.** Integrar sistemas existentes conectando bases de datos y la mejor clase de paquetes de soluciones en procesos de negocio flexibles.
- **Automatizar actividades rutinarias.** Se ejecutarán y optimizarán procesos de negocio automáticos.
- **Gestionar todas las fases de los procesos.** Ayudarán a descubrir, diseñar, desplegar, operar y analizar los procesos de negocio, dentro de un entorno integrado que soporte las necesidades de los gestores, de los ingenieros de procesos, de los departamentos funcionales y de los empleados.
- **Desplegar procesos.** Permitirán ser diseñados “on-line” por los usuarios de negocio y por los ingenieros de procesos juntos.
- **Proporcionar visibilidad y control total.** Permitirán a los procesos ser concebidos, desplegados, optimizados y analizados completamente, a través de múltiples aplicaciones. Proporcionarán una visión y control global de la organización completa.

Las tecnologías que un BPMS cubre e integra son, entre otras, las siguientes [5]: Sistemas de Workflow (flujos de trabajo), “Enterprise Application Integration” (EAI), servidores de aplicaciones, productos B2B, servicios web, etc.

Ya existen estudios que muestran las ventajas reales para las empresas cuando se utilizan un BPMS. De hecho, se ha comprobado que se puede reducir el tiempo de desarrollo de sistemas de información hasta un 75% y los costes de integración con otros sistemas hasta un 85%. Aunque los BPMS llevan muy poco tiempo en el mercado, están teniendo un impacto importante ya en muchas organizaciones y se espera que en un futuro lleguen a tener un papel tan significativo que el de los DBMS.

Un proceso es ejecutado por un BPMS y por los diferentes participantes en el proceso. El BPMS es el responsable de la coordinación de las transacciones definidas por el proceso, de manejar las instancias de los procesos, y de procesar las transacciones distribuidas [5].

Las transacciones de negocio (como un pedido de compra) y las transacciones de sistemas (como una transacción procesada en una tabla de una base de datos) pueden ser definidas mediante un proceso. Normalmente, las transacciones de negocio implican dos o más participantes, mientras que las transacciones de sistemas pueden implicar múltiples sistemas (transacciones distribuidas).

2. ESTÁNDARES DE PROCESOS DE NEGOCIO

En este capítulo se mostrarán los distintos estándares existentes actualmente para abordar el modelado y la implementación de procesos de negocio.

2.1. BPMN (Business Process Modelling Notation)

El objetivo principal de BPMN es ofrecer una notación entendible por todos los participantes en los procesos de negocio y su automatización, desde los analistas de negocio que crean los primeros borradores de los procesos, hasta los desarrolladores responsables de implementar la tecnología que lleva a cabo los procesos [8].

BPMN ha sido diseñado para ser fácil de usar y de entender, pero también proporciona la capacidad de modelar procesos de negocio complejos. También ha sido diseñado teniendo en cuenta la tecnología de Servicios Web.

Debido a la solidez matemática con que esta notación ha sido desarrollada (su semántica operacional está basada en el pi-cálculo), su transformación al lenguaje BPML (se explicará más adelante) se puede hacer directamente, de la misma forma que un modelo de datos se puede transformar directamente a un Lenguaje de Definición de Datos (DDL) [9].

BPMN sirve para especificar diagramas de procesos de negocio (DPN). Un DPN modela un flujo de procesos de negocio, indicando los eventos que ocurren al comenzar el proceso, las actividades que son llevadas a cabo y los resultados finales del flujo de proceso. Las decisiones de negocio y las ramificaciones de los flujos son modeladas usando pasarelas (*gateways*).

Además, el flujo un proceso puede contener subprocesos, los cuales pueden ser mostrados gráficamente mediante otro diagrama de procesos de negocio. Si un proceso no se descompone en subprocesos, es considerado como una tarea.

Mediante BPMN también se puede modelar “quién hace qué”, simplemente colocando los eventos y los procesos dentro de áreas sombreadas, llamadas piscinas (*pools*), la cuales especifican quien está llevando a cabo el proceso. Esas piscinas, además, pueden ser divididas en calles (*lanes*), las cuales representan normalmente los distintos departamentos o unidades de una organización, mientras que la piscina representa a la organización entera.

En la figura 5 se muestra un ejemplo de diagrama de procesos de negocio, especificado con el lenguaje BPMN.

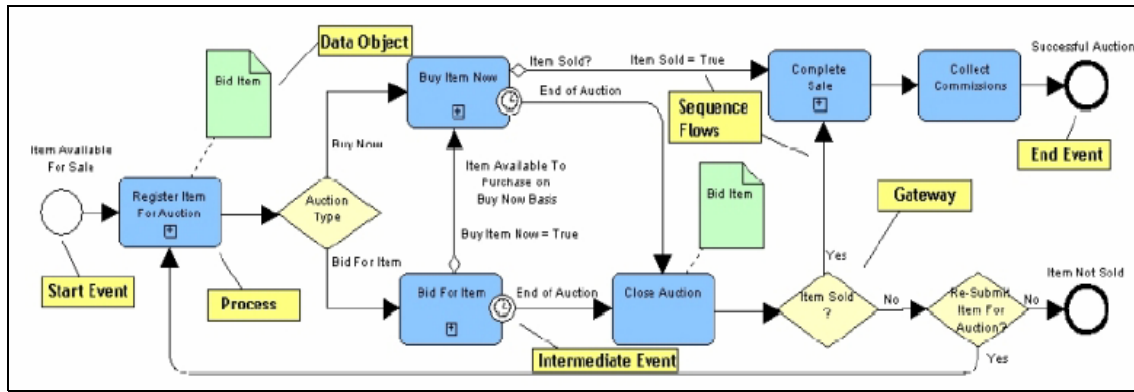


FIGURA 5. Ejemplo de diagrama de procesos de negocio usando BPMN [9].

2.2. BPML (Business Process Modelling Language)

BPML es un metalenguaje para el modelado de procesos de negocio, de la misma manera que XML es un metalenguaje para el modelado de datos de negocio. BPML proporciona un modelo de ejecución abstracto para procesos de negocio colaborativos y transaccionales basados en el concepto de una máquina de estados finitos transaccional.

BPML puede manejar participantes de diferentes clases. Desde sistemas gestores de bases de datos y componentes software hasta usuarios y socios de negocio (clientes y proveedores).

2.3. BPDM (Business Process Definition Metamodel)

OMG tiene su propio plan de desarrollo para abarcar el modelado de procesos de negocio. Este plan está incluido dentro de MDA, que ya ha sido presentada.

Debido a BPDM es un metamodelo dentro de la arquitectura MDA para implementar BPM tiene un papel clave para los intereses de este informe técnico. Por ello, BPDM se explica en detalle en una sección próxima como clave en la aplicación del paradigma MDA a la gestión de procesos de negocio.

2.4. SBVR (Semantics of Business Vocabulary and Business Rules)

Desarrollada por OMG, esta especificación define el vocabulario y las reglas para documentar la semántica del vocabulario de negocio, hechos de negocio y reglas de negocio. También define el esquema XMI para el intercambio de vocabularios de negocio y reglas de negocio entre organizaciones y entre herramientas software.

La especificación es aplicable para todo tipo de actividades de negocio de todos los tipos de organizaciones. Al ser de un nivel conceptual, es más adecuada para la gente de negocios que para el procesamiento automático de reglas, y está diseñada para ser usada para propósitos de negocios, independientemente de los diseños de sistemas de información [10].

SBVR es la especificación de un metamodelo para capturar expresiones en un lenguaje natural controlado y para representarlas en expresiones de la lógica formal. Con SBVR se espera conseguir la integración semántica de los negocios y la especificación formal de los requisitos de los sistemas en lenguaje natural. La lógica soportada por SBVR proporciona unas capacidades amplias de expresión para escribir especificaciones, incluyendo lógica de predicados de primer orden, lógica de orden superior, lógica modal, teoría de conjuntos y matemáticas.

SBVR es un metamodelo MOF, por lo tanto puede ser almacenado en un repositorio MOF, intercambiado y enlazado con otros metamodelos basados en MOF. SBVR está siendo integrado de forma plena en la arquitectura MDA de OMG.

2.5. BPEL4WS (Business Process Execution Language for Web Services)

BPEL4WS es una notación para especificar el comportamiento de procesos de negocio implementados mediante tecnología de Servicios Web. Permite modelar los procesos de negocio de dos maneras:

- a) Mediante procesos de negocio ejecutables se modela el comportamiento de un participante en una interacción de negocio.
- b) Mediante protocolos de negocio, por el contrario, se usan descripciones de procesos que especifican el comportamiento del intercambio visible de mensajes de cada parte involucrada en el protocolo, sin revelar su comportamiento interno. Las descripciones de procesos para protocolos de negocios son llamados procesos abstractos.

BPEL4WS proporciona un lenguaje para la especificación formal de procesos de negocio y protocolos de interacción de negocios [11].

BPEL4WS define procesos de negocio usando un lenguaje basado en XML. No define una representación gráfica de los procesos ni proporciona ninguna metodología particular para diseñar procesos.

Mediante BPEL4WS se define una orquestación de Servicios Web que son usados tanto por los procesos abstractos como por los ejecutables.

En la figura 6 se puede observar un fragmento de código BPEL4WS.

3. MDE APLICADO A BPM

En este capítulo se muestran las distintas propuestas existentes del paradigma MDE a BPM, así como las claves para la aplicación de este paradigma en cuanto a los estándares de modelado existentes.

Debido a que todas las propuestas *model-driven* aplicadas a BPM tienen como protagonista el paradigma MDA, en adelante este documento sólo hablará de MDA (teniendo en cuenta que MDA es un ejemplo concreto de MDE).

```

<process name="ncname" targetNamespace="uri"
  queryLanguage="anyURI"?
  expressionLanguage="anyURI"?
  suppressJoinFailure="yes|no"?
  enableInstanceCompensation="yes|no"?
  abstractProcess="yes|no"?
  xmlns="http://schemas.xmlsoap.org/ws/2003/03/business-
process/">
  <partnerLinks?>
    <!-- Note: At least one role must be specified. -->
    <partnerLink name="ncname" partnerLinkType="qname"
      myRole="ncname"? partnerRole="ncname"?>+
  </partnerLink>
</partnerLinks>
<partners?>
  <partner name="ncname">+
    <partnerLink name="ncname"/>+
  </partner>
</partners>
<variables?>
  <variable name="ncname" messageType="qname"?
    type="qname"? element="qname"?/>+
</variables>
<correlationSets?>
  <correlationSet name="ncname" properties="qname-list"/>+
</correlationSets>
.....
</process>

```

FIGURA 6. Fragmento de código BPEL4WS para definir un proceso, usando notación XML.

3.1. Propuestas de aplicación de MDA a BPM

A continuación se exponen distintas propuestas encontradas en la literatura analizada, las cuales presentan diversas alternativas de uso del paradigma MDA en la Gestión de Procesos de Negocio.

3.1.1. MDA como nexo de unión de los procesos de negocio entre varias organizaciones.

Los procesos de negocio, en ocasiones, no son internos a una organización, sino que requieren la interacción entre distintas organizaciones para llevar a cabo la cadena de valor completa que satisfaga las necesidades de dichas organizaciones. Esta categoría de procesos de negocio recibe el nombre de procesos de negocio colaborativos.

Las metodologías para desarrollar procesos de negocio colaborativos también tienen que tener en cuenta las metodologías usadas por las organizaciones participantes en dichos procesos. Esto implica el proceso de desarrollo de procesos de negocio, así como el modelado de conceptos usados para representar los procesos de negocio [12].

MDA se propone como el paradigma en que se basa el método de desarrollo de los procesos de negocio. Todas las organizaciones participantes tienen que modelar sus modelos independientes de la computación (CIM), independientes de la plataforma (PIM) y específicos de la plataforma (PSM) para sus procesos de negocio (figura 7).

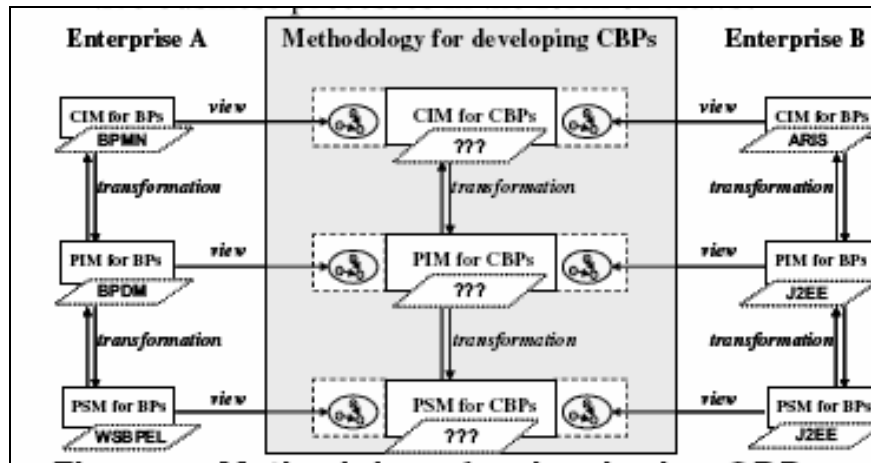


FIGURA 7. Representación de los procesos de negocio colaborativos usando la arquitectura MDA [12].

Para desarrollar procesos de negocio colaborativos es necesario que la información acerca de las interfaces externas de las organizaciones participantes esté disponible a los modelos de los procesos de negocio colaborativos. Esta información la proporcionan las vistas QVT (QVT views).

Una vez descrita la metodología de desarrollo de los procesos se plantean algunas cuestiones. Por ejemplo, se puede dar el caso en que haya dos organizaciones participantes en los procesos de negocio colaborativos. La primera usa para su CIM el lenguaje de modelado BPMN y la segunda usa el lenguaje ARIS. Las cuestiones son del tipo:

- ¿Qué lenguaje de modelado sería el apropiado para modelar los procesos colaborativos de negocio al nivel de CIM?
- ¿Qué conceptos del modelo BPMN y del modelo ARIS tienen que ser proporcionados por los participantes al CIM para los procesos de negocio colaborativos en las vistas (QVT views)?

Las mismas cuestiones surgirían a nivel de PIM y de PSM.

A nivel de CIM, el nivel de abstracción y los conceptos relativos a procesos de negocio son las características que deben definir a un lenguaje de modelado de procesos de negocio. ARIS y BPMN proporcionan ambos aspectos, aunque BPMN parece el más adecuado para ser usado a nivel de CIM. Las vistas de ambos modelos proporcionan una representación pública de los procesos, que deben ser compatibles entre sí para poder representar adecuadamente los procesos de negocio colaborativos.

Para elegir los lenguajes de modelado a nivel de PIM y PSM se siguen exactamente los mismos criterios que para el CIM. Así, se debe elegir un lenguaje común que pueda utilizar las vistas públicas o interfaces de los procesos de negocio de ambas organizaciones, dependiendo del lenguaje de modelado que esté usando cada una (figura 8).

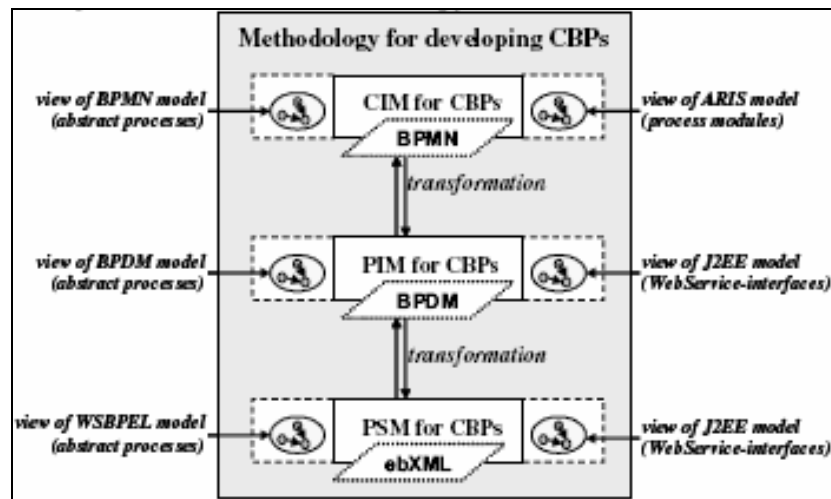


FIGURA 8. Representación de los procesos de negocio colaborativos usando la arquitectura MDA, una vez elegidos los estándares adecuados [12].

3.1.2. MDA como soporte al ciclo de vida en BPM

MDA y BPM tiene ciertas similitudes. BPM usa los modelos para automatizar la gestión de procesos de negocio y MDA usa los modelos formales para describir y automatizar el desarrollo y la integración del software. Ambos utilizan los modelos esforzándose por conseguir la máxima independencia de la tecnología.

Ahora bien, es necesario distinguir entre el modelado de procesos de negocio sin más y la gestión de procesos de negocio. La clave de esta distinción radica en el uso de los modelos de procesos. Antes de la gestión de procesos de negocio, un modelo de un proceso era simplemente un artefacto del proceso de diseño, creado para ayudar a los gestores de negocio a diseñar y comprender los procesos. Con la gestión de procesos de negocio, los modelos de procesos son artefactos de producción que dirigen varias aplicaciones de gestión de procesos, y deben ser lo suficientemente formales como para jugar ese papel [6].

El mayor valor de los productos BPM es la capacidad de permitir a los analistas de negocio modificar la manera en que un proceso es ejecutado sin tener que reprogramar el sistema o los sistemas que lo llevan a cabo. Pero realmente es muy difícil realizar transformaciones directamente, sin ningún paradigma que las guíe, entre modelos y código que implemente dichos modelos. Para solucionar esa dificultad, se sugiere que la mayoría de los lenguajes de notación y modelado de procesos de negocio proporcionen una transformación hacia el metamodelo de OMG para procesos de negocio (BPDM), y así permitir a las herramientas MDA generar cualquier código software necesario (figura 9) [13].

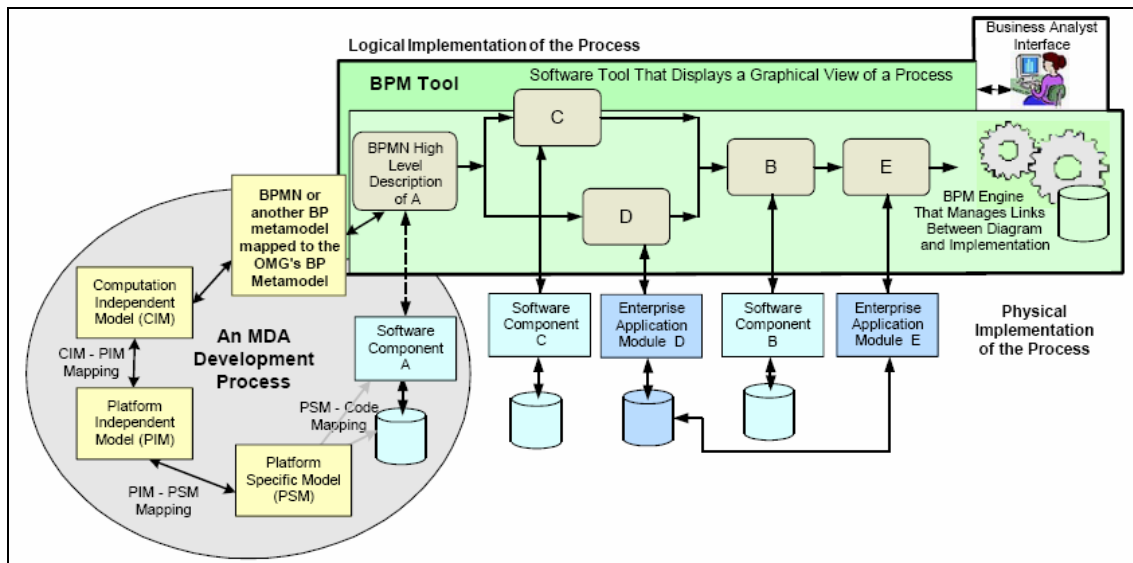


FIGURA 9. Proceso de modificación de un proceso de negocio usando MDA.

El modelo de más alto nivel de abstracción en MDA es el Modelo Independiente de la Computación (CIM). Un desarrollador de software podría imaginar que un diagrama de casos de uso UML fuera un modelo CIM. Pero ese tipo de diagramas no son populares entre gestores de negocios. En su lugar, esta gente prefiere diagramas organizacionales y de procesos los cuales también tienen cabida en el paradigma MDA, siempre que sean instancias de un metamodelo MOF. En otras palabras, MDA proporciona una estructura conceptual que se extiende desde los diagramas usados por los analistas de negocios hasta los usados por los desarrolladores de software y ofrece una organización conjunta, de manera que los requisitos expresados en unos modelos puedan ser trazados hacia otros diagramas de más detalle, derivados de los primeros.

Como se observa, MDA puede ser la mayor herramienta disponible para alinear los sistemas y tecnologías de la información con los procesos de negocio o viceversa, según se considere respectivamente desde el punto de vista del ingeniero/implementador de sistemas/software o desde la perspectiva del responsable/analista de negocio.

Siguiendo la arquitectura MDA, el modelo CIM sería la descripción de los procesos de negocio de la organización, utilizando un lenguaje de modelado comprensible por los analistas de negocio. En el caso de que dicho modelo no fuera instancia de un metamodelo MOF, debería ser transformado al metamodelo MOF para procesos de negocio (BPDM). A partir de transformaciones, este modelo CIM se transformaría en un modelo PIM, independiente de la plataforma que podría estar expresado en un lenguaje de modelado de sistemas de información como UML. A partir de aquí, el proceso de desarrollo MDA seguiría su curso normal, transformando hacia el modelo de plataforma específica (PSM) y el código que implementa dichos procesos.

Cuando se considera que ya existen los Sistemas de Gestión de Procesos de Negocio (BPMS), con motores que interpretan lenguajes específicos de dichos procesos, como BPML o BPEL4WS, el papel de MDA cambia respecto a la explicación anterior. En esta situación, las transformaciones se limitarían a la transformación de BPMN o cualquier otro lenguaje de modelado de procesos de negocio al metamodelo MOF BPDM. De esta forma, se tendría un modelo independiente de plataforma, el cual podría ser transformado a cualquiera de los lenguajes anteriores, sin necesidad de construir un modelo dependiente de plataforma.

Sabiendo que actualmente existen transformaciones entre el lenguaje de modelado BPMN y los lenguajes BPEL y BPML, se podría deducir que MDA sobra en este escenario. Sin embargo, al incorporar MDA a este proceso de transformación e interponer en las transformaciones el metamodelo BPDM, se incorpora la ventaja del intercambio de modelos MOF mediante el estándar XMI y la posible utilización de repositorios de modelos MOF, para almacenar los distintos modelos que una organización ha ido construyendo conforme a sus procesos de negocio, lo cual parece indispensable en estos tiempos.

Teniendo en cuenta todo lo anterior, el ambiente cambiante de los procesos de negocio de una organización se afronta con mucha más garantía mediante el uso de MDA. Así, simplemente, el analista de negocio tendría que modificar los CIMs que representan los procesos de la empresa y, mediante la aplicación de las transformaciones MDA se llegaría a la producción de código fuente de una manera más rápida, facilitando una convergencia total (por primera vez en la historia) entre la gente de negocios y los ingenieros del software.

3.2. BPDM. El estándar de MDA para BPM

Como ya se ha comentado, para que el paradigma MDA pueda ser incorporado a la gestión de procesos de negocio hace falta un nexo de unión. Ese nexo es un metamodelo estándar que forme parte de la familia de metamodelos MOF. Por ello, OMG está creando el metamodelo BPDM (Business Process Definition Metamodel) [14], del cual aun no ha salido la primera versión oficial.

BPDM es una descripción semántica de las relaciones lógicas entre varios elementos de alguna posible descripción de un proceso de negocio. No es una notación. Simplemente describe relaciones lógicas.

Una vez que el BPDM de OMG esté completado, cualquier compañía podrá transformar su metamodelo específico al metamodelo de OMG, y así usar XMI para transportar información acerca de los modelos desarrollados en una herramienta a otra herramienta que también conozca dicho metamodelo. De la misma manera, el estándar BPMN o los diagramas de actividad de UML 2.0 podrán ser transformados al estándar BPDM (figura 10).

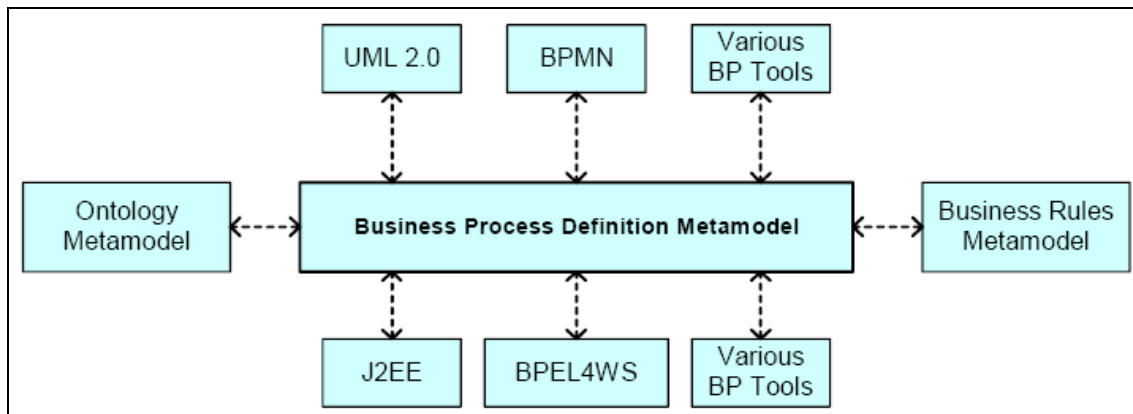


FIGURA 10. Visión del metamodelo BPDM respecto al resto de metamodelos [13].

OMG (creador de BPDM) y el equipo de BPMI (creador de BPMN) están trabajando juntos, y aseguran que la versión que salga a la luz de BPDM soportará todos los conceptos que es capaz de modelar BPMN.

IBM propone un UML-Profile de BPMN como notación para cualquier modelo que sea instancia del metamodelo BPDM. En cambio, OMG ha propuesto UML-Profile para los diagramas de actividad como lenguaje de notación de BPDM. En este último caso, sería necesario definir una transformación entre BPMN y esa notación definida por OMG.

3.3. Estándares BPM para los distintos tipos de modelos MDA

Como ya se ha explicado con anterioridad, el paradigma MDA propone la utilización de distintos modelos para distintos puntos de vista del sistema (CIM, PIM y PSM). Para el caso de la Gestión de Procesos de Negocio existen varias propuestas de utilización de lenguajes de modelado para cada tipo de modelo.

Por un lado se propone la utilización del estándar BPMN en lo más alto de los modelos MDA, es decir, en el CIM. A partir de él se propone aplicar las transformaciones necesarias para derivar el modelo UML equivalente y el código que implementa dicho modelo (figura 11) [13].

Por otro lado, existe una propuesta completa de lenguajes relacionados con los procesos de negocio y su posible aplicación en las distintas etapas de desarrollo de MDA [12]. En la tabla 1, se pueden observar tanto el modelo MDA donde se puede aplicar, como otros datos relativos a su notación, la organización o grupo que lo ha creado, y si es soportado por alguna herramienta.

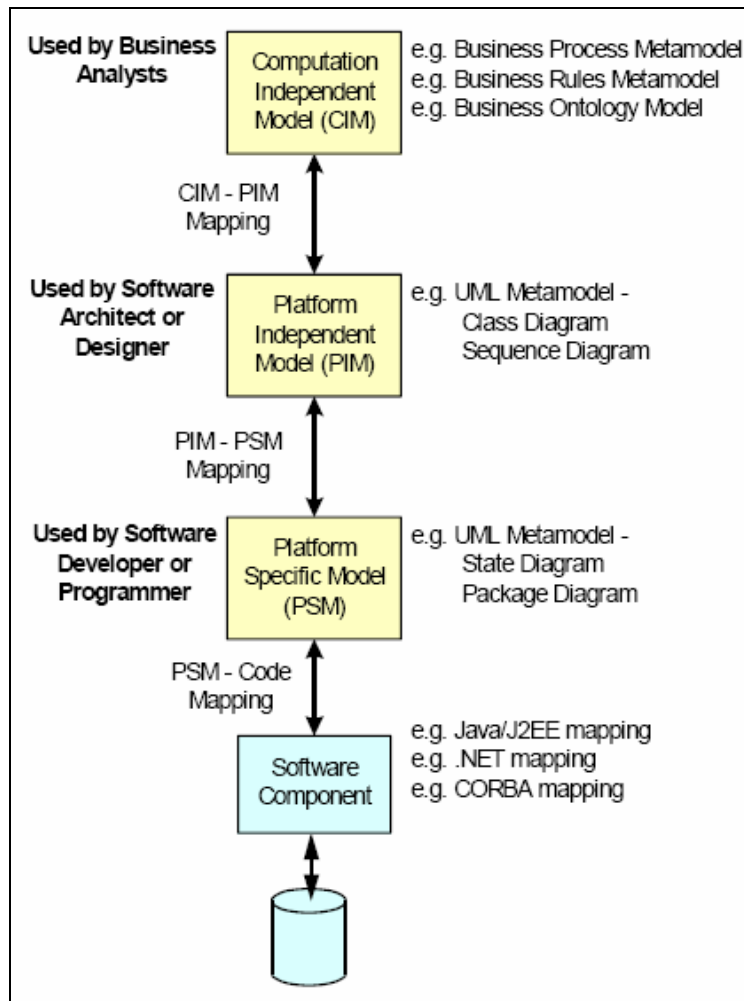


FIGURA 11. Utilización de MDA en el proceso de desarrollo BPM.

	Nivel de abstracción	Notación	Estandarización	Herramientas
AIRIS	CIM	Proporciona notación gráfica	---	AIRIS-toolset
BPDM	PIM	No proporciona notación gráfica. Posible notación gráfica con UML	OMG (en progreso)	---
BPML	PSM	Notación textual	BPMI.org	Algunas implementaciones
BPMN	CIM	Es una notación gráfica	BPMI.org	Alrededor de 20 implementaciones
ebXML	PSM	Notación textual.	UN/CEFACT OASIS	Varias implementaciones
BPEL4WS	PSM	Notación textual. Posible notación gráfica con UML-Profile	OASIS	Varias implementaciones
WS-CDL	PSM	Notación textual	W3C	---
J2EE	PIM, PSM	Posible notación gráfica con UML-Profile	Sun	J2EE es la base de muchas sistemas de aplicaciones Web

Tabla 1. Clasificación de los estándares de BPM según varios parámetros.

3.4. Limitaciones

Aunque parece que ambas filosofías juntas, BPM y MDA, podrían sacar beneficios la una de la otra, existen voces en contra de dicho acercamiento.

Aunque MDA y BPM son arquitecturas para sistemas software de organizaciones, la comparación entre ambos acaba aquí. A la vez que MDA es diseñado para permitir la generación de aplicaciones y componentes dirigidos por computador, el propósito de BPM es todo lo contrario, evitar escribir o generar más software, creando una nueva forma de trabajo, el proceso, que puede ser manipulado a una escala previamente inimaginable.

El propósito de BPM no es hacer más fácil la creación de más software. BPM es una arquitectura dirigida por diseños (DDA) y no una arquitectura dirigida por modelos (MDA). Los modelos BPML son directamente ejecutables sin necesidad de ninguna transformación. Aunque existen propuestas de “UML ejecutable”, hay diferencias fundamentales con BPM, además de diferencias entre ejecutar software y ejecutar procesos.

En MDA, y en la ingeniería del software en general, las capas de abstracción son usadas para permitir la transformación de software de una forma a otra. En BPM, mientras es posible estructurar y hacer capas de abstracción sobre los procesos, no hay un paso requerido entre el proceso de diseño y el de despliegue. Esto es debido a que los BPMS se ajustan a un modelo de procesos inmediatamente, igual que un RDBMS se ajusta a un modelo de datos [15].

Estas limitaciones son simplemente la opinión de un autor. Es decir, que aunque aquí se presenten argumentos en contra del acercamiento entre los paradigmas MDA y BPM, también existen muchas opiniones a favor. De cualquier modo, serán el tiempo y el avance en las investigaciones, los que darán la respuesta del éxito o fracaso de la aplicación del paradigma MDE a la Gestión de Procesos de Negocio.

REFERENCIAS

1. Bézivin, J., *MDA: From Hype to Hope, and Reality*, Conferenciante invitado en UML'03 (2003).
2. Schmidt, D.C., *Model-Driven Engineering*, IEEE Computer Society, Febrero (2006), pp. 25 – 31.
3. OMG: *MDA Guide Version 1.0.1*, OMG, 2003.
4. Hammer, M., Champy, J., *Reengineering the Corporation*. New York; HarperCollins, (1993)
5. Smith, H., Neal, D., Ferrara, L., Hayden, F., *The Emergence of Business Process Management*, CSC's Research Services, Enero (2002).
6. Frankel, D., *The Rise of Model-Driven Enterprise Systems*, Business Process Trends, Junio (2003), pp. 1 – 15.
7. Smith, H., *What a BPMS Is*, Business Process Trends, Febrero (2005).
8. OMG: *Business Process Modeling Notation Specification*, OMG, Febrero (2006).
9. Owen, M., Raj, J., *BPMN and Business Process Management*, Popkin Software, 2003
10. OMG. *Semantics of Business Vocabulary and Business Rules Specification*. OMG, Marzo (2006).
11. Andrews T., *Business Process Execution Language for Web Services*. Version 1.1. Mayo (2003).
12. Roser, S., Bauer, B., *A Categorization of Collaborative Business Process Modelling Techniques*, Proceedings of the 2005 7th IEEE International Conference on E-Commerce Technology Workshops (CECW'05), 2005.
13. Harmon, P., *The OMG's Model Driven Architecture and BPM*, Business Process Trends, Mayo (2004), pp.1 – 12.
14. OMG: *Business Process Definition Metamodel*, Request For Proposal, OMG, Agosto (2003).
15. Smith, H., *BPM and MDA: Competitors, Alternatives or Complementary*, Business Process Trends, Julio (2003), pp.1 – 15.

ACRÓNIMOS

BPDM	Business Process Definition Metamodel
BPEL4WS	Business Process Execution Language for Web Services
BPM	Business Process Management
BPMI	Business Process Management Initiative
BPML	Business Process Management Language
BPMN	Business Process Modeling Notation
BPMS	Business Process Management System
CIM	Computation Independent Model
DSML	Domain Specific Modeling Language
EAI	Enterprise Application Integration
ERP	Enterprise Resource Planning
MDA	Model Driven Architecture
MDE	Model Driven Engineering
MOF	MetaObject Facility
OCL	Object Constraint Language
OMG	Object Management Group
PIM	Platform Independent Model
PSM	Platform Specific Model
QVT	Query View Transformation
RDBMS	Relational Data Base Management System
SBVR	Semantics of Business Vocabulary and Business Rules
SPEM	Software Process Engineering Metamodel
UML	Unified Modeling Language
XML	Extensible Markup Language
XMI	XML Metadata Interchange