

Análisis y comparación de plataformas BRMS a través de una prueba de concepto

Gonzalo Martin Ibarra¹, Mg. Patricia Bazán²

¹ Facultad de Informática UNLP ² LINTI Facultad de Informática UNLP

ibarragm@gmail.com, pbaz@ada.info.unlp.edu.ar

Resumen

En la última década la implementación de los sistemas BRMS (*Business Rules Management System*) viene creciendo a medida que el negocio requiere más flexibilidad y velocidad a la hora de un cambio de reglas de negocio.

BRMS permite la administración y ejecución de reglas de negocio de distintas formas, de modo que no sea necesario la implementación con “código duro” y que las mismas sean escritas por los usuarios de negocio sin la participación del equipo técnico. Es importante mencionar el tema de costos ya que las implementaciones tradicionales requieren de esfuerzo en diferentes áreas y como consecuencia un mayor costo en tiempo y dinero.

Si bien desde el concepto mencionado en el párrafo anterior se puede inferir como un simple cambio o mejora en la creación e implementación de las reglas de negocio, podemos afirmar que en la realidad no es así ya que hay varios factores o características diferentes en ejecutar un proyecto basado en reglas de negocio que un proyecto de desarrollo de aplicaciones tradicionales.

Pensar en implementar un BRMS implica conocer que la metodología de desarrollo de los proyectos es diferente a la tradicional, que existen diferentes actores y roles para la ejecución de los proyectos, que es necesario un fuerte compromiso desde el área técnica y de negocio de las empresas, también en la adopción de una herramienta de implementación BRMS entre otros temas a destacar.

El área técnica deberá comprender que el ciclo de vida de un proyecto difiere de un proyecto de implementación de una aplicación tradicional y también el área de negocio deberá comprender que cumple un papel fundamental en el éxito del proyecto, siglas como BRE (*Business Rules Engine*), SBVR (*Semantics of Business Vocabulary and Business Rules*), actores como *rule architect*, *rule analyst*, documentos como “*Decision Point Table*”, “*Rule Template*” serán moneda corriente en una metodología de desarrollo BRMS.

Existen diferentes escenarios y también plataformas de aplicación para la ejecución de

reglas de negocio. Inicialmente podemos mencionar escenarios de aplicación en plataforma Java, servidores centrales (mainframes, donde se ejecutan programas COBOL), entornos SOA (*Services Oriented Architecture*) también sobre plataforma .Net.

Entendiendo que BRMS comprende la utilización de plataformas, herramientas y metodologías diferentes, no tan exploradas como las tradicionales y con fuerte indicio de crecimiento sostenido se decidió obtener más conocimiento investigando las diferentes opciones del mercado y además realizando una prueba de concepto con alguna de las plataformas más conocidas del mercado [1] [2] [3][8].

Palabras clave: BRMS (*Business Rules Management System*), BRE (*Business Rules Engine*), SBVR (*Semantics of Business Vocabulary and Business Rules*).

Contexto

El presente es un trabajo de fin de carrera de Licenciatura en Sistemas de la Facultad de Informática de la UNLP, del alumno Gonzalo Martin Ibarra, dirigida por la Mg. Patricia Bazán.

Introducción

La investigación realizada hasta el momento nos permite abordar y mencionar las características y los aspectos generales de la implementación general de un BRMS, definiendo los ítems a comparar luego de la prueba de concepto.

a. Clasificación de las reglas de negocio

Existen diferentes tipos de declaraciones que clasifican a las reglas de negocio, sin embargo no existe una declaración, definición o tipificación estándar de reglas.

Una clasificación general es categorizar las reglas en Términos, Hechos y Reglas. Un **término** es un sustantivo o una proposición con una definición acordada, Un **hecho** es una declaración que conecta términos, a través de preposiciones y frases verbales las cuales forman una regla de negocio y una **regla** es una

declaración que aplica lógica o cálculo de valores de la información.

Una regla se utiliza para el descubrimiento de nueva información o de una decisión acerca de tomar acción [3].

b. Metodologías de implementación

Los sistemas tradicionales son abordados por diferentes metodologías como RUP, XP, SCRUM, solo por nombrar algunas de las más conocidas, los cuales presentan diferentes etapas y documentos a elaborar, sin embargo no son parte de este estudio y entendemos que los lectores de este artículo poseen conocimientos sobre las mismas.

Los sistemas BRMS pueden ser aplicados en diferentes escenarios de implementación, es decir, en un sistema ya implementado donde es necesario realizar una reingeniería con el fin de extraer las reglas (*botton-up*) o bien un sistema implementado obteniendo las reglas de negocio realizando relevamientos con los usuarios de negocio (*top-down*).

A continuación presentamos las diferentes actividades y fases de la metodología ABRD (*Agile Business Rule Development*) las cuales se detallan en la Figura 1.

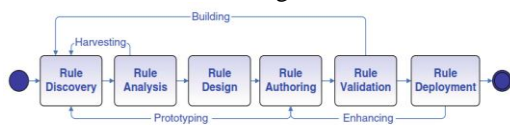


Figura 1. Metodología ABRD [1]

Rule Discovery: también llamado *Business Modeling* en la industria, tiene como objetivo desarrollar artefactos simples de modelado como descripciones de reglas de negocio, diagramas de entidad, y los mapas de procesos de negocio.

Rule Analysis: el objetivo de la actividad es entender el significado de la regla expresada por la persona de negocios y expertos en la materia y de quitar cualquier ambigüedad y la cuestión semántica o de interpretación.

Rule Design: en esta actividad se elaboran diferentes documentos para plasmar las reglas de negocios y comenzar a estrechar la brecha entre el análisis y la implementación de las reglas. Existen herramientas que facilitan el diseño de las reglas.

Rule Authoring: es la actividad de creación de las reglas de modo que puedan ser interpretadas por el motor de reglas. Si realizamos una analogía con las metodologías tradicionales sería la actividad de programación de la aplicación.

Rule Validation: esta etapa corresponde al test de las reglas en diferentes ambientes y facilidades que presentan las plataformas de BRMS.

Rule Deployment: es la actividad de "desplegar" las reglas en los diferentes ambientes definidos por el proyecto [4][10].

c. Arquitectura

Existen diferentes plataformas y arquitecturas, se presenta la arquitectura de IBM JRules como muestra la figura 2.

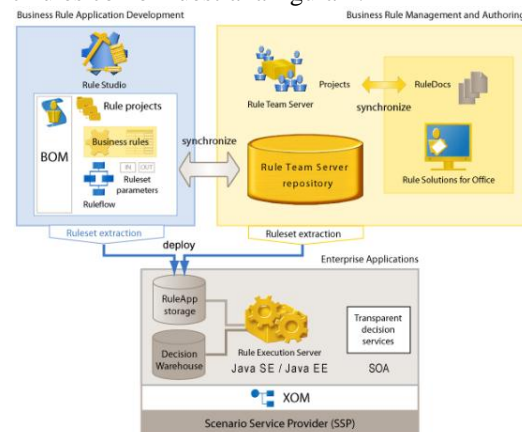


Figura 2. Arquitectura IBM JRules [1]

Es necesario un ambiente de trabajo para crear las reglas, en este caso sería el *Rule Studio* y también necesitamos un componente que interprete o ejecute las reglas, entonces el *Rule Execution Server* nos presta el servicio. Por último, el *Rule Team Server* permite conservar y manipular las reglas ya definidas.

La arquitectura JRules nos brinda dos componentes adicionales, *Decision Validation Services*, que básicamente se utiliza para realizar test y también *Rules Solution for Office* que es una herramienta pensada para facilitar la integración de reglas con componentes Microsoft.

Rule Studio: es el entorno de desarrollo para aplicaciones de reglas de negocio. Permite diseñar, crear, revisar y depurar reglas de negocio. Se integra en Eclipse. Los desarrolladores pueden tomar ventaja de esta

integración para desarrollar sus proyectos Java con los proyectos de reglas.. *Rule Studio* también dispone de herramientas para mantener las reglas sincronizados con el repositorio de *Rule Team Server*

Rule Execution Server: el motor de reglas es un componente utilizado para aislar y aplicar la lógica de negocio (las reglas) de una aplicación

Rule Team Server: es un servidor y repositorio que dispone de una herramienta web colaborativa para la creación, gestión, validación e implementación de reglas de negocio. Proporciona un sistema de almacenamiento centralizado de reglas de negocio y sus metadatos.

Decision Validation Services: es un módulo que los desarrolladores pueden utilizar para probar y simular las reglas de negocio con respecto a los escenarios, que representan casos reales o ficticios.

Rule Solutions for Office: es un entorno de creación de reglas en Microsoft Office. Las reglas pueden ser escritas utilizando Microsoft Word, las tablas de decisión se editan utilizando Microsoft Excel.[1][9]

d. Roles

La metodología comentada anteriormente implica la participación de roles especiales que no son parte de una metodología tradicional de desarrollo de aplicaciones. Dichos roles son:

Arquitecto de Reglas: es responsable de diseñar la infraestructura necesaria para editar reglas, definir el repositorio de reglas y trabajar en conjunto con el resto del equipo para desarrollar una aplicación basada en reglas.

Desarrollador: es responsable del desarrollo, prueba, depuración y despliegue de aplicaciones de reglas de negocio. Interactúa con los modelos de objetos, APIs, y el entorno de desarrollo (Java EE servidores de aplicación o Java SE).

Analista de Reglas ("Rule Analyst"): es el equivalente a un "analista de negocios" o un "analista funcional" si lo comparamos con los roles de las metodologías de trabajo tradicionales. Debe poseer un fuerte conocimiento del negocio como así también

estar familiarizado con las aplicaciones basadas en reglas, conocer el funcionamiento de un motor de reglas y el proceso de descubrimiento y análisis de reglas de negocio.

Escritor de Reglas ("Rule Writer"): asiste a los analistas de negocio en la creación de las reglas tal como las requieren el motor de reglas.

Administrador de Reglas ("Rule Administrator"): gestiona la creación y despliegue de reglas, garantizando la integridad y ciclo de vida de las mismas [1][7].

d. Motor de Reglas de Negocio

La utilización de motores de reglas es aplicable en aquellas situaciones en que la lógica requerida (por ejemplo, reglas) es lo suficientemente dinámica para hacer que la inclusión de tal lógica en el código fuente del software resulta poco práctico, por lo tanto es conveniente llevar las reglas a un repositorio de reglas externalizado, es decir, fuera de la aplicación que las utiliza y contar con un intérprete de las mismas.

En el nivel más básico, un motor de reglas se compone de tres componentes, una base de conocimiento-(se trata de normas representados en un formato computacional), y un motor de inferencia o ejecución que puede razonar acerca de los datos de entrada basados en el contenido de los conocimientos-de base con el fin de generar algún tipo de salida, tal como un conjunto de instrucciones o de alerta, el tercer componente, es la facilidad de la ingeniería del conocimiento, que proporciona la capacidad de conservar el contenido de la base de conocimientos sobre una base de datos

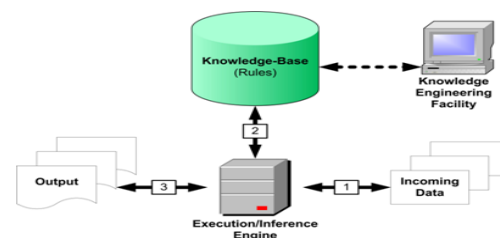


Figura 3 Arquitectura Motor de Reglas [4]

En la figura 3 se refleja la entrada de información (1) la cual es analizada por el intérprete (con los diferentes algoritmos) o motor de inferencia de reglas en conjunto con la base de conocimiento (2) y esta a su vez interactuando con la base de datos de reglas, luego de este proceso se llega a la salida de información (3).

Los algoritmos más conocidos son Rete, Secuencial y Fastpath los cuales poseen

diferentes características y es factible que los motores de reglas los combinen de acuerdo a la regla de negocio a interpretar [1][4][7][8].

e. Características

Un BRMS tiene las siguientes características y responsabilidades:

- El almacenamiento y el mantenimiento de un repositorio de reglas de negocio que representan las políticas y procedimientos de una empresa.

- Mantener estas reglas (la lógica de negocio) separados de los componentes de integración necesarios para los sistemas de computación distribuida. Esta característica se puede relacionar a los sistemas tradicionales con enfoque BPM (Business Process Management) en el sentido de utilizar las reglas de negocio en un ambiente distribuido y pensando los procesos desde el punto de vista de negocio

- La integración con las aplicaciones empresariales, de modo que las reglas se pueden utilizar para todos los negocios la toma de decisiones, utilizando los datos normales de negocios.

- Normas que forman en conjuntos de reglas independientes pero con conexión en cadena y las inferencias que realizan dentro de tales conjuntos de reglas

- Permitir que los analistas de negocio e incluso los usuarios para crear, entender y mantener las normas y políticas de la empresa con el mínimo de aprendizaje que requieren. Esta característica es bien diferente a los sistemas tradicionales ya que ubica al usuario en un rol preponderante en todo el ciclo de vida de la aplicación ya que el usuario de negocio participa desde la definición hasta el mantenimiento de las reglas de negocio, no siendo el caso en los sistemas tradicionales donde participa en el análisis y quizás en las pruebas de aceptación.

- La automatización y la facilitación de procesos de negocio.

- Creación de aplicaciones inteligentes que interactúan con los usuarios a través de diálogos naturales, comprensible y lógica, Esta característica quizás es el ideal de los sistemas orientados a reglas; para lograr el objetivo se requiere de madurez de las reglas como así también un fuerte conocimiento del negocio por parte de los usuarios[1][2][9].

LÍNEAS DE INVESTIGACIÓN Y DESARROLLO

Esta línea de investigación realiza un análisis profundo de las características de las plataformas y herramientas involucradas en el

desarrollo de proyectos basados en la administración de reglas de negocio. Para lograr dicho objetivo se seleccionaron dos plataformas:

- *WebSphere ILOG JRules*
- *Drools*

Para poder realizar la comparación de las características se elaboró una prueba de concepto que consiste básicamente en la implementación de requerimientos de negocio a través de ambas herramientas.

Es válido mencionar que para replicar o simular el proceso completo de la creación de una aplicación BRMS es necesario contar con diferentes personas, documentos, una empresa real que tenga una necesidad de negocio real y el tiempo suficiente para desarrollar el proyecto, desde el descubrimiento de las reglas de negocio hasta la implementación de las misma [2][9].

RESULTADOS Y OBJETIVOS

De acuerdo a la disponibilidad de recursos y objetivo de la prueba de concepto se decidió utilizar un documento o "Fuente original" elaborado de modo que sea posible investigar diferentes características de las plataformas, para que de esta forma se pueda dar inicio al proceso de desarrollo y que este sea el disparador para la obtención de las reglas de negocio, asimismo para poder desarrollar el caso de prueba fue necesaria la participación en distintos roles los cuales fueron anteriormente descriptos .

Luego de la aplicación de la prueba de concepto se realizará la comparación de la plataforma utilizando la Tabla 1.

Gran parte de las características a comparar fueron obtenidas de referencias mencionadas sobre este artículo e incluyendo algunos aspectos no mencionados.

La tabla 1 fue dividida en 4 secciones, Atributos Generales donde se incluye el desarrollo de los BRMS en Argentina midiendo su base instalada y recursos profesionales además de comparar precios y facilidades de instalación y soporte. Se incluye el IDE (*Integrated Development Environment*) el cual nos permite crear, diseñar y mantener reglas de negocio, a mi entender es muy importante ya que una interface amigable nos permite suavizar el cambio en la forma de pensar en ejecutar proyecto y alivia la "resistencia al cambio"; luego la sección *rule engine*, en donde incluimos los tres algoritmos de inferencia y también el soporte de *debugging*, muy importante también para dar herramientas que faciliten el testeado de reglas (recordemos que estamos ante un escenario diferente de pruebas del sistema tradicional), por último la sección de integración, importantísimo desde el punto de

vista de SOA, BPM y la necesidad de las grandes empresas de integrarse con diferentes sistemas [5][6].

Tabla 1 Comparación BRMS [5]

Atributos Generales	
Precio	
Facilidad de Instalación	
Soporte Técnico	
Servicios Profesionales disponibles (Argentina)	
Metodología de implementación	
Adherencia a la industria	
Soporte al ciclo completo de implementación	
Base instalada en Argentina	
IDE(Integrated Developed Environment)/Templates	
Disponibilidad de templates/ejemplos a reutilizar	
Soporte para formato de expresiones	
Soporte para tablas de decisión	
Referencia cruzada de reglas	
Control de versiones	
Chequeo de ambigüedad	
Rule Engine	
Soporte a RETE	
Soporte al Algoritmo Secuencial	
Soporte a Fastpath	
Soporte para debugging	
Despliegue de reglas durante la ejecución ("hot deploy")	
Capacidad para administrar un alto volumen de reglas	
Facilidad de administración y estructuración de reglas	
Capacidad para planificar deploys	
Performance / Escalabilidad	
Integración	
Facilidad de integración con herramientas de Middleware	
Integración con Webservices	
Integración con ambientes J2EE	
Integración con COBOL	
Integración con .Net	
Integración con BPM	

FORMACIÓN DE RECURSOS HUMANOS

BRMS es una nueva forma de abordar los proyectos que requiere un ciclo de vida diferente a la implementación tradicional de las aplicaciones. Es necesario también que las implementaciones se realicen en empresas que se encuentren preparadas tecnológicamente y que dispongan de los recursos humanos adecuados para desarrollar esta forma de trabajar. Entendemos que los proyectos de BRMS no están difundidos en la Argentina, sin embargo si se puede detectar un fuerte crecimiento en la industria norteamericana de software. Conociendo esta situación real se decidió colaborar y aportar un granito de arena en la difusión de esta alternativa de implementación de aplicaciones.

REFERENCIAS

- [1] Jerome Boyer, Hafedh Mili, "Agile Business Rule Development".
- [2] Stephen D. Hendrick, Kathleen E. Hendrick, "The Business Value of Business Rule Management Systems".
- [3] Barbara Von Halle " Business Rules Applied".
- [4] Joel Saltz, Joyce Niland, Philip Payne, Hemant Shah ,Douglas Stahl "Rules Engine Technologies Across caBIG Workspaces "
- [5] Ian Graham, " Service Oriented Business Rules Management Systems".
- [6] Daniele Di Bona, Giuseppe Lo Re, Giovanni Aiello, Adriano Tamburo, Marco Alessi, "A Methodology for Graphical Modeling of Business Rules".
- [7] Xiaoming Feng, Mani Subramanian, "Incorporating Business Rule Engine Technology in Control Center Applications"
- [8] Senlin Liang Paul Fodor Hui Wan Michael Kifer, "OpenRuleBench: An Analysis of the Performance of Rule Engines".
- [9] Saqib Ali, Ben Soh, Torab Torabi "Using Software Engineering Principles to Develop Reusable Business Rules".
- [10] Matthew L. Nelson, Robert L. Rariden, Ravi Sen, "A Lifecycle Approach towards Business Rules Management".