

Reglas de negocio: Clasificaciones, lenguajes e implementaciones

Colectivo de Autores

Edición: Liset Ravelo Romero

Corrección: Estrella Pardo Rodríguez

Martha Beatriz Boggiano Castillo, Ramiro Pérez Vázquez, Luisa Manuela González
González, 2014

Editorial Feijóo, 2014

ISBN: 978-959-250-940-5



Editorial Samuel Feijóo, Universidad Central “Marta Abreu” de Las Villas, Carretera a
Camajuaní, km 5 ½, Santa Clara, Villa Clara, Cuba. CP 54830

RESUMEN

Las reglas de negocio, sus clasificaciones, lenguajes y formas para expresarlas, así como las maneras de implementarlas constituyen temas de interés actual para el desarrollo de los Sistemas de Información (SI). Este interés está motivado por las bondades que propicia el tratamiento explícito de las reglas, como son la disminución del costo de programación y el aumento de la eficacia de dichos sistemas.

Esta monografía presenta un estudio acerca de las reglas de negocio, a partir de conjuntos de categorías usados por diferentes autores en los que se observa que todos proponen categorías de reglas que involucran de una manera u otra a los datos del negocio, generalmente almacenados en bases de datos.

Se muestran diferentes niveles de expresión para las reglas de negocio, así como formalismos y lenguajes utilizados. También se analizan distintas maneras de implementar las mismas, tanto por parte de los programadores, como por herramientas que las generan de forma automática. Se destacan aspectos sobre la implementación de reglas de negocio mediante recursos de bases de datos y finalmente, se tratan puntos de vista sobre elementos a tener en cuenta para la modificación consecuente de ellas.

ÍNDICE

Páginas

Resumen	3
Introducción	6
1.1 Las reglas de negocio.....	8
1.1.1 Características fundamentales de las reglas de negocio.	10
1.2 Categorías para reglas de negocios.	11
2 Capítulo 2: Maneras de expresar las reglas de negocio y lenguajes para escribirlas.	16
2.1 Niveles y formas de expresar para reglas de negocios.....	16
2.2 Lenguajes para reglas basados en patrones.	17
2.2.1 Uso de la notación punto.	17
2.2.2 Lenguaje OCL	18
2.3 Lenguajes para reglas basados en XML: RuleML, SWL, RIF, OMG-PRR.	19
2.3.1 RuleML.....	20
2.3.2 SWRL.....	21
2.3.3 RIF Rules interchange format.	22
2.3.4 Especificación SBVR. Conceptos fundamentales.	22
2.3.5 Comparación lenguajes más usados.....	24
3 Capítulo 3: Implementación de las reglas de negocio.	25
3.1 Formas de implementar reglas de negocio.	25
3.1.1 Sentencias de programas.....	26
3.1.1.1 Problemas en la gestión de las reglas de negocio codificadas en el programa. 26	
3.1.2 Secuencia de Comandos (Scripts).....	27
3.1.3 Componentes	27
3.1.4 Mecanismos de bases de datos	28
3.1.4.1 Reglas de negocio y restricciones de integridad en bases de datos. 28	
3.1.4.2 Restricciones (Constraints) CHECK:	29
3.1.4.3 Disparadores (triggers).....	30
3.1.5 Los procedimientos almacenados.....	31
3.1.6 Vistas.....	32
3.1.7 Los motores de reglas de negocio.....	32
3.1.8 Los motores de reglas de inferencia.	33
3.1.8.1 Mecanismos de inferencia.	35
3.1.9 Sistemas de flujos de trabajo.....	35
3.1.10 Tablas de búsqueda.....	36
3.2 Tendencias en la gestión de reglas de negocio.	37

3.2.1	Justificación para implementar las reglas de negocios en bases de datos.	38
3.2.1.1	Operaciones CRUD y reglas de negocios.....	39
3.2.2	Consideraciones para la implementación automática de reglas de negocios.	40
3.3	Repositorio de reglas.....	41
3.4	Influencia de las modificaciones de las reglas en los SI.....	41
3.4.1	Control de cambios y versiones de reglas de negocio	42
3.4.2	Versiones e historial de reglas de negocio.....	43
4	Conclusiones.	45
5	Referencias.....	46
6	Anexos.....	50
Anexo 1	50

INTRODUCCIÓN

Desde de la década de los 80 se trabaja en la búsqueda de formas adecuadas de escribir explícitamente las reglas de negocio (RNs) para lograr ciertos niveles de independencia entre las reglas en sí y la implementación de los sistemas de información (SI). La investigación sobre reglas de negocio se extiende también a buscar estrategias y modos de trabajo para lograr implementaciones de reglas generadas automáticamente.

Se han desarrollado herramientas comerciales para facilitar las implementaciones de las reglas, los más utilizados son los motores de reglas de RNs de inferencia, que trabajan con reglas de la forma *if.. then*.

También las RNs tienen un importante papel en el mundo de las bases de datos, se utilizan provechosamente en el diseño de las bases de datos; pues la extracción de las RNs es una actividad clave para la creación del esquema conceptual (García González, 2009).

Existen RNs que no son fáciles de representar en el diagrama de diseño conceptual, ni en el diagrama del diseño lógico porque son restricciones y sucesos que abarcan más especificaciones que los representables en el diseño estructural de la base de datos. Estas reglas pueden también estar centradas en los datos del negocio y por esto, probablemente queden implementadas con mayor naturalidad dentro de bases de datos. Se observa que las reglas ligadas a restricciones sobre los datos, generalmente tienen que ver con las funcionalidades CRUD (del inglés, create, read, update, delete) (Gottesdiener, 1997b), porque inciden en los eventos de creación o inserción, lectura, actualización y eliminación de los datos.

El estudio de las RNs se facilita si las mismas son clasificadas acorde a algún criterio que se desee ponderar. Surgen así diferentes clasificaciones, unas desde la propia perspectiva del negocio y otras desde la perspectiva de los SI; incluso también es posible clasificarlas desde una perspectiva unificada (Date, 2001); Besembel and Chacón, 2001; Morgan, 2002d; Ashwell, 2006; Lowenthal, 2005; Ross, 1997; Von Halle and Ronald, 2002; Matei, 2006; Zoet *et al.*, 2011).

Frecuentemente los autores proponen un conjunto de categorías (clasificación) para facilitar el estudio de las reglas junto a formalismos o lenguajes para expresarlas, muestra de ello son: RuleSpeak (Ross, 2009), los patrones de Morgan (Morgan, 2002d), y las categorías Weiden (Weiden *et al.*, 2002).

Estos formalismos o lenguajes pueden hacerse complejos y la mayoría se caracteriza por utilizar los términos del negocio desde una perspectiva conceptual. Existe también el criterio de que la escritura de las reglas debe expresarse teniendo en cuenta

diferentes niveles, desde el lenguaje informal del experto, hasta el lenguaje de programación ejecutable en una computadora.

Las implementaciones de las reglas pueden ser acometidas de diversas maneras, algunas han sido muy usadas por los programadores de SI escribiendo directamente el código, actualmente las formas de implementarlas tienden a la automatización.

El objetivo de este trabajo es estudiar diferentes clasificaciones de RNs, formalismos, lenguajes para expresarlas y las maneras de implementarlas, haciendo especial hincapié en aquellas categorías de reglas reconocidas por diferentes autores en relación con los datos del negocio.

La monografía se estructura en tres capítulos:

En el capítulo 1 se exponen algunas definiciones de RNs, propiedades y características de las reglas, se presentan conjuntos de categorías de reglas de diferentes autores, y se enfatiza en aquellas categorías que se relacionan con los datos del negocio.

En el capítulo 2 se estudian diferentes niveles para escribir las reglas, así como lenguajes y formalismos que existen para escribirlas incluyendo los que se destacan para lograr la interoperabilidad de las reglas en la Web.

En el capítulo 3 se analizan las maneras de implementar las RNs, se enfatiza el estudio en los motores de reglas y en los recursos con que cuentan los gestores para reflejar, sobre todo aquellas analizadas como restricciones de integridad.

Capítulo 1: Las reglas de negocio y sus categorías

En este capítulo se presentan diferentes definiciones de RNs, así como las categorías de reglas descritas por algunos de ellos. Se observa cómo en la mayoría de los conjuntos de categorías (clasificaciones de reglas) se reconocen algunas que están estrechamente relacionadas con los datos del negocio, que comúnmente se almacenan en bases de datos relacionales.

1.1 Las reglas de negocio

Una RNs según Ronald G. Ross, conocido como el padre de las RNs, (Ross, 2010) es una regla que está bajo jurisdicción del negocio, lo cual significa que estas pueden ser creadas, revisadas y eliminadas cuando el negocio lo estime conveniente.

Las RNs son ciudadanos de primera clase en el mundo de los requisitos (Business Rules Group, 2003). Los requisitos, reflejo de las políticas o RNs, han sido tradicionalmente implementados por el programador como parte del código fuente de las aplicaciones, o en los objetos de alguna base de datos, o en ambas; con un consiguiente costo de mantenimiento.

Desde mediados de la década del 80 las RNs comienzan a tratarse con especial distinción (Nelson *et al.*, 2009). Se observa una tendencia a gestionar las reglas de forma sistemática y centralizada, así se establecen sus precedentes históricos en la realización de aplicaciones de intérpretes simples basados en reglas, usando el algoritmo RETE iniciador de los productos comerciales de máquinas de RNs a partir 1986: System Corporations G”, 1986, Hley Systems; 1989; ILOG SA; 1988, Blaze Advisor.

Los precedentes históricos de tratar las RNs con especial distinción se refieren a la década de los ochenta, (Nelson *et al.*, 2009). Con tendencia a gestionar las reglas de forma sistemática y centralizada, se desarrollan aplicaciones de intérpretes simples usando el algoritmo RETE y así se inician los productos comerciales de máquinas de RNs: System Corporations G” y Hley Systems en 1986, Blaze Advisor de 1988 , así como ILOG SA de 1989.

Según Hay y Healy (2000), “una regla de negocio es una sentencia que define o restringe algunos aspectos del negocio. Tiene la finalidad de establecer la estructura del negocio, controlar o influenciar el comportamiento de negocio”... Generalmente son estudiadas desde dos perspectivas: desde la perspectiva del negocio, que pertenece a cualquier restricción de la conducta de las personas en la empresa; desde la perspectiva de los (SI) que está relacionada con los hechos que son grabados como datos y las restricciones sobre los cambios a los valores de tales hechos.

Las RNs constiuidas como sentencias permiten a los usuarios expertos definir políticas, condiciones, modelar el conocimiento del negocio en unidades pequeñas y aisladas de otros requerimientos. (Morgan, 2002a).

El término de enfoque de RNs (BRA, acrónimo del inglés: Business Rules Approach) se encuentra con frecuencia para referirse al manejo de las RNs independientemente de las aplicaciones que las hacen cumplir; de acuerdo con este enfoque, las RNs son relativamente independientes de los requerimientos de un SI y necesitan un tratamiento especial (Zimbrão *et al.*, 2003).

Morgan (2002d y 2002a) enumera un conjunto de propiedades deseables en las declaraciones de RNs, las cuales se manifiestan en cualquier lenguaje o dominio de aplicación:

Atómica: no pueden ser descompuestas sin que se pierda información.

No ambigua: tienen solamente una interpretación obvia.

Compacta: típicamente son frases cortas.

Consistente: juntas, ellas proporcionan una única y coherente descripción.

Compatible: usan las mismas condiciones en el resto del modelo de negocio.

Varios autores coinciden en que las RNs tienen un significativo efecto sobre la adaptabilidad y flexibilidad de las aplicaciones (Bajec and Krisper, 2005; Ross, 2003; Morgan, 2002c; Zachman, 1987).

Se reconocen varios beneficios derivados del uso de las RNs, según Lowenthal (Lowenthal, 2005) los tres más importantes son agilidad, reducción de costo y transparencia.

Debido a la diversidad y complejidad de las RNs los autores tienden a agruparlas y clasificarlas siguiendo diferentes perspectivas, siempre con el objetivo común de organizar todas las reglas del análisis de requisitos del negocio.

En ocasiones los propietarios de negocios y los desarrolladores no coinciden en la manera de tratar las reglas (Gottesdiener, 1997b). Desde el punto de vista de los SI se debe establecer cuál es el origen de las reglas: (1) traducciones de las RNs en sus implementaciones y (2) creación de nuevas reglas provenientes del diseño del sistema. Ross (2006) denomina a estas últimas: reglas de sistema, para distinguirlas de las RNs propiamente.

Las clasificaciones (conjunto de categorías) que combinan las cuestiones empresariales con las características técnicas del sistema tienden a obstaculizar la trazabilidad entre los SI y las necesidades del negocio, que puede dar lugar a paradojas que son difíciles de conciliar (Carver and Morgan, 2012).

Se observa que al definir conjuntos de categorías de reglas, variados autores definen subconjuntos de éstas vinculadas a los datos que se manejan en el negocio, lo que

puede ser común porque los datos y los sistemas de bases de datos son repositorios centrales de información para los negocios (Ataullah and Tompa, 2011).

1.1.1 Características fundamentales de las reglas de negocio

Morgan (2002d) se refiere a un conjunto de características de las RNs que se consideran fundamentales, y se describen a continuación.

Las RNs, en términos generales, son restricciones: ellas definen condiciones que deben ser verdaderas en situaciones especificadas, *bajo las cuales un proceso es realizado o las nuevas condiciones que existirán después de que un proceso ha sido completado*. (La palabra proceso es usado aquí en el sentido general.). No deben considerarse como descripciones de un proceso o procesamiento>;

Las RNs definen lo que debe ser: más bien el *qué*, que *cómo* va a ser. Un juego de reglas que definen pre y post condiciones puede actuar como una especificación para un proceso, sin hacer alusión los mecanismos por los cuales las condiciones previas son transformadas en las post condiciones.

Las declaraciones de regla de negocio en el modelo de negocio definen la lógica deseada del negocio. Describen una situación que el negocio exige.

Si se expresan como funciones booleanas, las RNs siempre devolverían un valor de verdadero; desde una perspectiva lógica de regla de negocio, no hay excepciones. Una excepción supuesta a una regla de negocio es simplemente otra regla de negocio. Algunas reglas pueden parecer obvias, pero si es así, está bien ya que se muestra que se ha construido una lógica de negocio comprensible.

El analista de negocio debe especificar una serie de declaraciones claras sobre la lógica base de un negocio. El aspecto de la claridad, es crucial y deben ser tan claras que el experto o dueño del negocio debe aceptar inmediatamente como válido o rechazar como inválido. El analista debe comprender que la gente del negocio no entienda de lógica de primer orden y que no tienen una perspectiva de tecnología, pero ellos pueden hacer que el negocio funcione usando solamente declaraciones simples, que el analista debe captar.

La experiencia ha mostrado en más de dos mil años que es posible hacer declaraciones claras y lógicas usando solamente palabras ordinarias en inglés, francés, chino, español, etc. La lengua no importa mientras la lógica esté clara.

El verdadero poder de las RNs según Morgan (2000) está en:

- La capacidad de hacer declaraciones de nivel de negocio que pueden ser traducidas de un modo casi directo a una parte de un sistema operacional.
- El efecto combinado de un número relativamente grande de declaraciones simples, de modo que juntas, tengan un impacto mayor que la suma de las partes individuales.

1.2 1.2 Categorías para reglas de negocios

Las diferentes clasificaciones de las reglas en conjuntos de categorías, se originan debido a los diversos puntos de vista para organizar las reglas: así se encuentran clasificaciones desde las perspectivas del negocio, desde los desarrolladores de SI, y clasificaciones que combinan ambas perspectivas.

A continuación se describe un estudio acerca de diferentes clasificaciones de RNs o conjuntos de categorías, se analizan aquellas comunes a varios autores, que se refieren a reglas, que surgidas en el negocio afectan el comportamiento de los datos del mismo y por tanto repercutirán en la representación digital de ellos, almacenados comúnmente en bases de datos. Para cada categoría analizada se exponen los autores que las defienden y las características generales de su funcionalidad.

Weiden (Weiden et al., 2002) se basa en las propiedades semánticas del negocio para clasificar RNs. Reconoce, entre otras, las categorías: estructura de conceptos, persistencia e historia. Las primeras se asocian a los conceptos del negocio (que son instanciados en los datos del negocio) y las relaciones existentes entre ellos las de persistencia e historia se vinculan a los datos del negocio que deben manejarse con características de persistencia.

Morgan (2002d) defiende que la forma más conveniente de expresar RNs es con un conjunto de patrones adecuados, pues además de ahorrar un tiempo valioso esto ayuda a hacer cumplir la consistencia de este enfoque. Los patrones se basan en las formas o tipos de reglas que se esperan encontrar y para ello es conveniente usar una clasificación que permita definir patrones, uno para cada tipo o categoría de RNs dentro de cada conjunto de categorías (clasificación).

Los cinco patrones propuestos por Morgan consideran de una manera u otra algún sujeto, sus características y hechos, así como la existencia de un modelo de hechos que reconozca términos y frases aceptadas para el dominio del negocio. Presenta los patrones de: Restricción básica, Lista de restricciones, clasificación, enumeración y cómputo.

Patrón de Restricción Básica: Es el más común entre los patrones para reglas de negocio, establece una restricción sobre el sujeto de una regla.

<determinante> <sujeto> [no] (debe | tiene) <característica>
[(si | a menos que) <hecho>].

Patrón de Lista de Restricción: Este patrón también restringe al sujeto, delimitando la cantidad de hechos verdaderos tomados de una lista.

<determinante> <sujeto> [no] (debe | tiene) <característica>
(si | a menos que) como mínimo <m> [no más de <n>] de las
siguientes es verdadera: <lista de hechos>.

Clasificación: Establece la definición para un término del negocio.

<determinante> <sujeto> [no] es definido como <clasificación>
[(si | a menos que) <hecho>].

Enumeración: Este patrón establece un rango de valores que pueden ser legítimamente tomados.

<determinante> <resultado> debe ser elegido de la siguiente
[abierto | cerrado] enumeración: <lista enumerada>.

Cómputo: Establece relaciones entre términos del negocio suficientes para hallar o establecer un valor.

<determinante> <resultado> es definido como <algoritmo>

<determinante> <resultado> = <algoritmo>. Entre las formas más comunes de clasificar RNs se encuentran aquellas relacionadas con las propiedades inherentes al negocio en sí, a los eventos relacionados con estos, así como al modelado de datos y a los SI asociados.

Muchos autores coinciden en algunas categorías de reglas y disienten en otras.

Las reglas llamadas de restricción son incluidas por Date (Date, 2001) Solivares (Besembel and Chacón, 2001) Morgan (2002a), Ashwell (2006), Lowenthal (2005), Ross (1997), Von Halle (Von Halle and Ronald, 2002), Ioana Matei (2006) y otros.

Solivares (Besembel and Chacón, 2001) apunta que en cierto modo las reglas de restricción se solapan con las reglas del modelo de datos, pues estas también impiden la introducción de datos erróneos, pero la diferencia estriba en que las reglas de restricción condicionan el valor de los atributos o propiedades de una entidad con más especificación que las restricciones básicas existentes sobre las mismas.

Matei (2006) reconoce las reglas de restricción como restricciones obligatorias, que especifican una declaración completa que debe ser verdadera, además reconoce las líneas directivas o restricciones suaves, que pueden ser interpretadas como sugerencias. La sugerencia es una declaración completa que expresa un mensaje acerca de una circunstancia que no debería ser verdadera o falsa si no es sólo una

advertencia y deja al humano la decisión de denegarlo o no. Tanto las restricciones obligatorias como las líneas directivas se agrupan en la categoría de las reglas de información de restricción sobre el acontecimiento comercial.

Andreescu (Andreescu and Mircea, 2009) presenta las reglas de enumeración para establecer un conjunto de posibles valores que un atributo de una entidad puede tomar, y las reglas de integridad para capturar algunas restricciones que se aplican a los atributos de la entidad. En definitiva ambas pueden ser enfocadas como reglas de restricción.

Según la concepción de SBVR, las reglas de acuerdo a su formulación lógica se clasifican en dos tipos (OMG, 2008a): reglas estructurales o de necesidad y reglas operacionales o de obligación.

De las categorías anteriores, las estructurales especifican las restricciones de integridad que no pueden ser violadas. Las restricciones de integridad están directamente relacionadas con la estructura de las entidades del negocio, representadas por los términos del negocio; que en última instancia conforman la estructura de los datos.

Las reglas de cómputo o cálculo son reconocidas por Ross (1997) como un subtipo de sus reglas de producción. Von Halle (Von Halle and Ronald, 2002) y Matei (2006) consideran la regla de cálculo como una de las categorías a tener en cuenta. Matei (2006) las localiza dentro del grupo que crea nueva información sobre la representación de los eventos del negocio.

Una regla de cómputo es una declaración completa que provee operaciones para lograr el valor de un término, donde los operadores pueden incluir suma, diferencia, producto, cociente, cuenta, mínimo, máximos y promedios (Von Halle, 2001). El resultado del cómputo puede ser un valor nuevo para un atributo.

Ejemplo:

La tasa mensual de préstamo es computada como la tasa anual de préstamo / 12 meses.

También se reconoce un conjunto de reglas como las consideradas por Ross como reglas de definición, usadas para organizar el conocimiento básico del negocio contribuyendo al significado de conceptos, y centrándose en la esencia del propio negocio (Ross, 2010). Asimismo, Ross llama reglas de derivación a las reglas que permiten hacer la definición de términos a partir de los existentes (Ross, 1997). En las categorías de reglas propuestas por Andreescu (Andreescu and Mircea, 2009) se presenta la categoría *if.. then*, se puede observar que estas en muchos casos permiten definir o clasificar un término del negocio como subclase de otro.

Por otra parte se aprecia en las categorías de RNs asumidas por Zoet (Zoet *et al.*, 2011), que usa algunas asociadas directamente a los datos, a estas reglas les llama reglas de condición de datos, y reglas de control de resultados.

Ante el estudio analítico de diversas categorías de reglas presentadas por diferentes autores, se puede observar que existen conjuntos de tipos de reglas que están directamente relacionadas con las instancias de los tipos de entidades del negocio y sus correspondientes propiedades.

Del análisis de las categorías de reglas de varios autores se observa que en no pocas de estas clasificaciones se presentan algunas categorías con objetivos comunes, que involucran directamente entidades del negocio y que tienen por tanto implicaciones en el manejo que hacen los SI sobre los datos.

La posición de Date (2000a) acerca de las RNs está centrada en los datos, manifiesta sus criterios a favor de que las reglas sean parte de las bases de datos; las reglas primariamente garantizan la integridad de la base de datos, las restricciones de rangos y también deben ser parte de la base de datos. Sin embargo, defiende las ventajas de mantener las reglas separadas del motor de bases de datos; existencia independiente de las reglas y de la base de datos (Date, 2000b, 2001), así las reglas pueden ser aplicadas a los datos almacenados por bases de datos como DB2, Oracle, Súbase, etc., pero lo más importante es que se mantengan almacenadas en un repositorio central, lo que permite que puedan ser mantenidas independientemente de los datos, aplicaciones e infraestructura. Las reglas centradas en los datos probablemente quedan implementadas más naturalmente dentro de la base de datos, pues pueden tener incidencia más directa sobre los datos y en las funcionalidades CRUD (create, read, update, delete) (Gottesdiener, 1997a) que son comunes a todos los gestores de datos.

La expresión “regla de negocio centrada en los datos” debe ser tratada con mucho cuidado, pues, por ejemplo, Morgan (2002d) define que aquellas reglas de la lógica del negocio que representan interrelaciones pudieran no registrarse dentro de las categorías de RNs. Ejemplo: *Un paciente cuenta con uno o más exámenes médicos*, (interrelación 1:M, entre la entidad “paciente” y “exámenes médicos”), según Morgan *no se trataría como una regla de negocio sino como un hecho del mismo* y, sin embargo, un planteamiento que adiciona una restricción a esto, sí se considerará como una RN: *Un paciente no puede contar con más de 50 exámenes físicos*.

Uno de los aspectos esenciales en el trabajo con las RNs y los SI es permitir la satisfacción de las necesidades de la organización con el sistema de manera que en el contexto de un sistema de información, se tenga en cuenta las cuestiones de

organización en particular y se pueda responder a la pregunta ¿cómo se aborda esto en el sistema de información? (Carver and Morgan, 2012).

Capítulo 2. Maneras de expresar las reglas de negocio y lenguajes para escribirlas

En este capítulo se analizan diferentes niveles de expresión de RNs presentadas por distintos autores, que coinciden en que estas sean expresadas en sus primeros niveles en lenguaje natural, pasando por un nivel más formal, que aún no es aceptado por la computadora, para ser traducidas a un nivel formal de lenguaje de computadora. Se presentan lenguajes basados en patrones, lenguajes basados en XML y el uso de la notación punto para expresar reglas de negocios.

2.1 Niveles y formas para expresar reglas de negocios

Algunos autores (Morgan, 2002^a; Von Halle and Ronald, 2002) coinciden en que, independientemente de la clasificación de la regla de negocio, estas tienen diferentes formas de expresión de acuerdo a diferentes niveles de abstracción, y que inicia con el lenguaje natural.

Son cuatro las formas de expresar las reglas (Von Halle and Ronald, 2002), cada una para una audiencia diferente: conversación informal del negocio, versión en lenguaje natural, versión en lenguaje de especificación y versión en lenguaje de implementación de reglas. Una regla comienza su vida en la conversación informal de las personas del negocio, luego, se realiza una versión más disciplinada en lenguaje natural cuya audiencia es la comunidad del negocio. El lenguaje natural presenta algunas insuficiencias, tales como falta de precisión y redundancia, por lo que se requiere que se expresen las reglas en un lenguaje que ya reúne todas las cualidades requeridas. Tal lenguaje es el de especificación de reglas. Es declarativo y disciplinado, y está dirigido tanto al personal del negocio como al técnico. Pero este lenguaje solo nos dice qué acometerse según la regla, no cómo. Por ello las reglas se traducen (o mudan) del lenguaje de especificación al lenguaje de implementación, el cual tiene todo el potencial para ser ejecutado.

Morgan (2002a) distinguen sólo tres formas de expresión de RNs: *informal*, con la RN como sentencia en lenguaje natural, tal y como el cliente del negocio desee; *técnico*, se combinan referencias a datos estructurados, operadores y restricciones con el lenguaje natural, nivel intermedio entre la escritura de la regla en lenguaje natural y su implementación; *formal*, nivel que proporciona sentencias conforme a una sintaxis definida y proporciona la funcionalidad automática de la regla.

En este trabajo se pondera la última clasificación, reconociendo que el nivel técnico es similar al de especificación, así como el formal al de implementación.

Existen diversidad de lenguajes para representar RNs, por lo general no se presentan de antemano clasificados en uno de estos niveles.

2.2 Lenguajes para reglas basados en patrones

Existen diversos lenguajes para escribir RNs en forma de patrones. Estos lenguajes usan frases claves para expresar las reglas de manera estructurada. Entre ellos se publican RuleSpeak (BRS, 2004), los patrones de Weiden (Weiden *et al.*, 2002), y los patrones de Morgan (2002d).

RuleSpeak® es un conjunto de pautas para expresar las RNs de forma concisa que facilita la redacción de la regla para cada una de las partes involucradas en el negocio; es desarrollado por Ross y publicado por el *Object Management Group* (OMG) en 2007.

Weiden, por su parte, propone diecisiete patrones de reglas, agrupados en tres categorías (Weiden, 2000): estructurales, de comportamiento y de administración. Estas categorías representan diferentes visiones del negocio: las concernientes a la descripción de aspectos estáticos del negocio, definición de las condiciones sobre la ejecución de tareas en el negocio y definición de las restricciones de alto nivel sobre el negocio.

Las categorías o tipos de reglas en forma de patrones presentadas por Morgan son cinco: restricción básica, lista de restricciones, enumeración, clasificación y cómputo. Morgan exige la representación posterior de la regla en un lenguaje de nivel técnico.

Las RNs expresadas en un nivel técnico deben ser escritas en un lenguaje capaz de permitir llevar una regla a un lenguaje formal, y sencillo de implementar. Esta idea es confirmada por varios autores como Zimbrão *et al.* (2003), Demuth, (2005); Heidenreich *et al.* (2008), Zimbrão *et al.* (2003).

2.2.1 Uso de la notación punto

La notación punto —del inglés *Dot Notation*— es un estilo más que un lenguaje formal. La programación orientada a objetos, la utiliza en la cualificación de los objetos. En (Morgan, 2002a) se propone el uso de tal notación para eliminar la ambigüedad del lenguaje natural a la hora de expresar las reglas del negocio. *Dot Notation*, familiar a los programadores, y su alternativa *Of-Notation* son propuestas de Halpin (Halpin, (2006) para expresar RNs donde es necesario representar las relaciones entre los conceptos, para ayudar en la navegación a través de las clases del modelo del negocio. Esto deriva en una mejor claridad en la expresión de reglas más complejas.

2.2.2 Lenguaje OCL

Varios son los lenguajes declarativos ya creados para expresar las reglas del negocio, que pudieran catalogarse como de nivel técnico. Estos lenguajes recogen los conceptos del negocio y los convierten en expresiones más o menos inteligibles para un especialista técnico. El propietario del negocio no puede lidiar con tal tipo de expresión, pero al diseñador se le humaniza el trabajo (Morgan, 2002d).

Al Lenguaje de Modelado Unificado (del inglés *Unified Modeling Language*, UML), estándar para la modelación de negocios y sistemas de software, se le incorpora en la versión 1.1, un lenguaje para la especificación de restricciones, el Lenguaje de Restricción de Objetos OCL (del inglés *Object Constraint Language*), véase Warmer (2003) (Warmer, (2003) para más detalles. El OCL es un lenguaje para expresar restricciones en modelos orientados a objetos. Con OCL se complementan los aspectos que el diagrama de UML no permite, decir al modelador cosas que los diagramas no sustentan, como las restricciones. También permite hacer aclaraciones a ambigüedades que pueden surgir cuando los diagramas de clase UML tratan de mostrar relaciones complejas. En los últimos años UML ha sido aceptado por la mayoría de los desarrolladores de software como herramienta de visualización, de documentación, de análisis y de diseño (Demuth, 2004). Con OCL pueden escribirse, para los modelos conceptuales descritos con UML, las restricciones que constituyan RNs. Un ejemplo, para la especificación del problema de negocio de un hospital infantil para restringir la edad de los pacientes a menos de 18 años es necesario declarar en OCL la expresión siguiente:

```
context Paciente
inv: self.edad < 18
```

Según indica Bruegge (Bruegge and Dutoit, 2009) y (Tedjasukmana, 2006; OMG, 2010) OCL puede ser usado con distintos propósitos: como lenguaje de navegación y de consulta, para especificar restricciones sobre operaciones, para describir pre y post condiciones de operaciones y métodos, así como otros. Realmente el papel de OCL se concentra en completar las especificaciones de los diagramas UML, como dice Morgan, ha puesto orden al UML (Morgan, 2002d). Es un lenguaje para definir restricciones de objetos para el diseño de clases con UML, con un conjunto de posibilidades amplias de funcionalidades reflejadas en herramientas CASE (Santos and Invernizzi, 2004). Sin embargo, Morgan en (Morgan, 2002d) no le da un peso especial al uso del OCL.

2.3 Lenguajes para reglas basados en XML: RuleML, SWL, RIF, OMG-PRR

La mayoría de las herramientas comerciales actuales para tratar reglas de negocio, son los denominados Sistemas de Gestión de Reglas de Negocio (SGRN), estos usan variedad de lenguajes propietarios para representar las reglas de producción que gestionan, y cada uno de estos sistemas tienen sus características específicas de trabajo. La estructura general de los lenguajes propietarios para representar reglas es la siguiente:

```
Paquetes de reglas
reglas
  regla nombre
    variables locales
  si
    antecedentes
  entonces
    consecuentes
  fin
  variables globales
  importaciones
```

A partir de estas representaciones, que pudieran clasificarse como de nivel técnico de acuerdo al epígrafe 1.1.2, no existe una estructura común para representarlos en un formato único internamente para la máquina.

Actualmente se aprecian esfuerzos de estandarización para representar estas reglas de la forma *if then* que garanticen el intercambio de reglas en la Web. Entre estos lenguajes se destacan RuleML (Rules Markup Language), (Boley *et al.*, 2010), el SWRL (*Semantic Web Rules Language*), RIF(*Rules Interchange Format*), (Martínez Hernández, 2010) PRR (*Production Rules Representation*), que han surgido en la búsqueda de un lenguaje común para motores de reglas con algoritmos de encadenamiento hacia adelante, para tratar las reglas expresadas como <if> <then>..... Estos lenguajes tuvieron sus inicios en la iniciativa CommonRules de IBM proseguidos por el BRML (*Business Rules Markup Language*) (BRML, 2002) y SRML (Simple Rules Markup Language) (Thorpe, 2001).

El SBVR (*Semantic Business Vocabulary and Rules*) (BRG,2009), (OMG, 2008b) proporciona una forma estándar para formar modelos de vocabulario, conceptos y formulaciones semánticas; no es un lenguaje que las personas del negocio utilicen para escribir reglas. Más bien proporciona una forma para modelar los significados del negocio que ha sido expresado en un lenguaje propio del personal de negocio. SBVR

no impone más requerimiento en el lenguaje del negocio que claridad. Bajo la guía de SBVR el personal del negocio no requiere conocimientos o habilidades informáticos para definir las reglas de su negocio (Baisley, 2008). No ofrece ni un modelo listo para ser transformado en software, ni un lenguaje de modelación a ser implementado.

A continuación se relacionan aspectos esenciales de algunos de los lenguajes más usados para escribir RNs, teniendo en cuenta las necesidades de interoperabilidad.

2.3.1 RuleML

Es una especificación para cubrir todos los aspectos de las reglas en la *Web* y su interoperación. Es una familia de lenguajes cuyos esquemas XML permiten un intercambio de reglas en la *Web* con alta precisión. Los dos tipos de reglas principales que se representan con RuleML son: las reglas de *deliberación* y las reglas de *reacción* (Boley *et al.*, 2010).

Las reglas de deliberación incluyen: reglas de derivación y modal, las que involucran hechos, solicitudes, restricciones de integridad y reglas de Horn (en datalog), además de reglas de negación clásica.

Las reglas de reacción incluyen: Procesos de eventos complejos (*Complex Event Processing*, CEP), Reglas de representación del conocimiento (*Knowledge Representation*, KR), Reglas de Evento-Condición-Acción (ECA), y Reglas de producción (CA). Así, RuleML puede combinar todas las partes de ambos tipos de reglas: deliberación y reacción.

RuleML Datalog constituye el núcleo o kernel de las reglas de derivación, construido a partir de la intersección de los lenguajes SQL y Prolog (Martínez Hernández, 2010), puede representar información relacional allí donde todos los atributos son frases en lenguaje natural; con este una relación puede ser n-aria, es decir, tener cualquier número fijo de atributos.

Un ejemplo de regla en RuleML: *Un cliente es premium si su gasto ha sido superior a 5000 euros en el año anterior.*

La representación en RuleML de la regla de negocio es como sigue:

```
<Implies>
  <then>
    <Atom>
      <Rel>premium</Rel>
    <Var>customer</Var>
```

```

</Atom>
<if>
  <body>
    <Atom>
      <Rel>gasto</Rel>
      <Var>cliente</Var>
      <Ind>superior a 5000 euros</Ind>
      <Ind>año anterior</Ind>
    </Atom>
  </if>
</Implies>

```

En esta regla se presentan como relaciones o tablas, Premium y Gasto. Para Premium, la variable customer y para Gasto los valores que deben igualar a dos variables de esta relación: los valores son “5000 euros” y “año anterior”. El sentido de las reglas en RuleML está en que si existe la causa X entonces el resultado es Y, o similar.

Esta iniciativa RuleML se ha convertido en un enlace con la especificación de reglas a través de la Web.

2.3.2 SWRL

SWRL es la combinación de los sublenguajes OWL DL, OWL Lite (del lenguaje OWL, *Web Ontology*), con el lenguaje RuleML. Se extiende con un conjunto de axiomas de reglas de Horn. Las reglas propuestas son de la forma de implicación entre un antecedente y un consecuente. Una regla de negocio que pueda enunciarse como una implicación lógica: antecedente (*body*) implica consecuente (*head*), puede representarse con formato SWRL.

Tanto el antecedente como el consecuente están formados por uno o más átomos. Un antecedente vacío implica verdad absoluta (*true*). Un consecuente vacío implica la falsedad del antecedente (*false*). Los átomos en estas reglas pueden ser de la forma: $C(x)$, $P(x,y)$, $sameAs(x,y)$ o $differentFrom(x,y)$, donde C es una descripción OWL, P es una propiedad OWL, y “x”, “y” son también variables individuales de OWL o valores de datos OWL. Se usan además relaciones funcionales.

Las reglas se pueden expresar en sintaxis abstracta o sintaxis legible (*Lage*). A continuación se presenta una regla en sintaxis legible.

Para expresar: Si un paciente tiene edad < 18 años entonces el paciente es un paciente pediátrico. Así su representación en sintaxis legible es la siguiente

```
Paciente(?x1)^tieneedad(?x1,?age1)^swrlb:lowerThan(?age1,18)→
PacientePediátrico(?x1)
```

Las variables son marcadas con un signo de '?' como prefijo. Las relaciones funcionales pueden escribirse de dos formas: `_ op:numeric-add(?x,3,?z)`, o

Para expresar: x1 tiene padre x2 y x2 tiene hermano x3 entonces x3 es tío de x1.

Representación SWRL:

```
tienePadre(?x1,?x2) □ tienehermano(?x2,?x3) □
tieneTío(?x1,?x3)
```

En una sintaxis abstracta, de SWRL se puede escribir, de la manera siguiente:

```
Implies(Antecedent(tienePadre(I-variable(x1) I-variable(x2))
tieneHermano(I-variable(x2) I-variable(x3)))
Consequent(tienetío(I-variable(x1) I-variable(x3))))
```

Este ejemplo también pudiera ser expresado en la sintaxis RDF/XML (Martínez Hernández, 2010).

Las ontologías ayudan al intercambio de datos y significados entre servicios Web. OWL se encuentra generalmente expresado en sintaxis RDF/XML.

Las tres expresiones de las reglas son equivalentes y toda regla representada en la forma legible podrá ser representada en la forma abstracta y en el formato RDF/XML.

2.3.3 RIF Rules interchange format

Es un lenguaje de intercambio de RNs definido por un grupo específico de trabajo del W3C. Pretende asegurar la interoperabilidad entre tecnologías basadas en reglas, permitiendo consistencia con respecto al resto de las especificaciones del w3c *OMG-PRR- (Production Rules Representation)*

Según las especificaciones de OMG 2007, explica el estándar de OMG para la representación de reglas de producción y cubre diferentes requisitos referenciados por Martínez Hernández (2010).

2.3.4 Especificación SBVR. Conceptos fundamentales

El metalenguaje SBVR (del inglés *Semantics for Business Vocabulary and Business Rules*) ha sido definido por el grupo dedicado al estudio de las RNs llamado *Business*

Rules Team, BRT (BRG, 2009). SBVR proporciona una forma estándar para formar modelos de vocabulario, conceptos y formulaciones semánticas; no es un lenguaje que las personas del negocio utilizan para escribir reglas. Proporciona una forma para modelar los significados del negocio que han sido expresados en un lenguaje propio del personal de negocio. SBVR impone al lenguaje del negocio el requerimiento de claridad.

Según SBVR, las reglas del negocio de acuerdo a su formulación lógica se clasifican en dos tipos:

- Reglas estructurales o de necesidad: especifican las restricciones estructurales (restricciones de integridad), no pueden ser violadas.
- Reglas operacionales o de obligación: regulan las operaciones del negocio, si son violadas llevarían a un estado no válido del negocio.

SBVR puede ser soportado por múltiples herramientas de diferentes fabricantes que proporcionan validación, análisis, alineación, combinación y composición de RNs.

SBVR cuenta con cinco características relevantes, las cuales constituyen su base, estos aspectos son: la comunidad, empresas para las cuales se establecen los términos de las RNs, con las subcomunidades con las que puede tener significados compartidos; la formulación lógica que permite la formulación lógica de la semántica con una sintaxis formal, abstracta, para capturar la semántica de un conjunto de significados compartidos; la representación del negocio en vocabularios.

Realmente no existe una explicación clara de cómo la categorización de reglas en SBVR asiste a la construcción y mantenimiento de SI en el aspecto práctico (Carver and Morgan). Tampoco existen explicaciones detalladas de cómo las distinciones conceptuales del metamodelo de SBVR se preservan en otros modelos de reglas, por ejemplo en (Group, July, 2000). Además en (Carver and Morgan) se hacen propuestas de que los profesionales de la información pueden preferir considerar esquemas de categorización alternativa, si la categorización es considerada importante, por ejemplo las reglas pudieran ser categorizadas para aquellas funciones de negocio en las que ellas impactan (cuentas, mercado, producción, etc.), por los roles de negocios que tienen autoridad para actualizar estas, por unidades de negocios y otros. Por supuesto la misma regla pudiera existir en múltiples categorías.

2.3.5 Comparación de los lenguajes más usados

Es interesante observar la complejidad con que se presentan los lenguajes estándares para representar las RNs y garantizar la interoperabilidad de estas, sobre todo en entornos Web.

A continuación se presenta la tabla 1, tomada de Martínez Hernández, (2010) que ilustra la comparación de los lenguajes estándares tratados anteriormente.

Tabla 1. Comparación de lenguajes para reglas de negocios

Lenguaje	Organismo de estandarización	Nivel de abstracción	Herramientas disponibles
RuleML	-	Medio	Implementaciones (jDrew, Mandarax)
SBVR	OMG	Alto (metalenguaje)	---
SWRL	W3c	Medio	Editores y razonadores (Protegé)
RIF	W3C	Medio	Varias implementaciones
OMG-PRR	OMG	Bajo	---

Algunos de estos lenguajes se presentan muy ligados al nivel de implementación y tienen un nivel de abstracción bajo, *OMG-PRR*, porque asumen determinados aspectos del entorno tecnológico. Un nivel de abstracción alto implica que no están relacionados con la implementación. El metalenguaje *SBVR* es completamente independiente de su implementación, puede creerse que resulta sencillo para los usuarios escribir, interpretar y comprender las RNs que se definen con el lenguaje definido a partir de dicho metalenguaje, pero en realidad puede ser muy complejo. Actualmente existen escasas herramientas para trabajar con él (aparte de las guías proporcionadas en el documento que define el estándar).

OMG PRR es de un nivel de abstracción bajo, porque asume que existe un modelo UML para el sistema en el que se van a implantar las RNs y que se está trabajando, por tanto, con un lenguaje de programación orientado a objetos.

Los lenguajes *SWRL* y *RIF*, están especialmente orientados a la integración con ontologías. Éstas recogen los conceptos a los que es posible hacer referencia en las RNs. Herramientas como *Protege* disponen de extensiones para la especificación de RNs expresadas en *SWRL*.

Capítulo 3: Implementación de las reglas de negocio

En este capítulo se describen las diferentes formas de implementar las RNs en los SI, las ventajas y desventajas de cada una de estas. Se analizan los problemas que se presentan cuando las reglas son mal tratadas dentro del código del sistema, y presentan maneras de lograr más flexibilidad a su tratamiento y se incluyen las tendencias en el desarrollo de tipos de herramientas para administrar las RNs. Se consideran aspectos que justifican la implementación de las reglas en las bases de datos relacionales y se aborda el *repositorio* de reglas, componente fundamental de las herramientas para gestionar las reglas. Por último, se evalúan elementos básicos que deben sustentar estas herramientas para permitir modificaciones de las reglas.

Formas de implementar reglas de negocio

Morgan (2002e) afirma que las RNs pueden estar en diferentes capas de la aplicación: en el cliente, en la capa de datos, en la capa media.

En algunos casos, la elección de la ubicación será determinada por las opciones disponibles: el diseñador tendrá que valorar las alternativas antes de elegir la forma en que una determinada regla se aplicará. El objetivo principal con la implementación es obtener el mejor ajuste a su entorno particular.

Existen formas muy disímiles de implementar las reglas, incluso pueden existir varias técnicas para implementar una misma regla (Morgan, 2002b). Al considerar cada una de las alternativas para determinar cuál funciona mejor en una determinada situación, se debe tener en cuenta:

- La viabilidad a largo plazo de la estrategia.
- Rendimiento en tiempo de ejecución.
- El grado de cumplimiento de la regla.
- La flexibilidad.
- La capacidad para mantener las operaciones del negocio.

Entre las formas de implementación (Morgan, 2002f) se destaca: código de programa a través de código en línea, secuencias de comandos, componente especializado de reglas, motores de reglas, mecanismos de bases de datos, los sistemas de flujo de trabajo *con un motor interno y estilos personalizados para las reglas de codificación*, tablas de búsqueda.

Cada una de estas alternativas presenta sus características, ventajas y desventajas. A continuación se explican brevemente:

Sentencias de programas

Esta ha sido la vía de aplicación más común. En cualquier lenguaje de programación se puede seleccionar entre ramas de código alternativo, basado en una condición dada, que es la manera clásica de programar una regla de negocio.

Una manera de hacer consistente esta forma es encapsular la regla o conjunto de reglas en una llamada de función, o en métodos de clases, etc.

Las *ventajas* de poner reglas directamente en el código son la sencillez y el rendimiento. Si se desea incluir algunas reglas de una actividad realizada con frecuencia o que tiene que llevarse a cabo dentro de limitaciones de tiempo ajustadas, la codificación directa de las reglas puede ser la opción más atractiva.

La *desventaja* de este enfoque es que un cambio en una regla requiere un cambio correspondiente en el código por lo que requiere programación, generalmente el lugar donde hay que hacer el cambio es difícil de encontrar. Bajec (Bajec et al., 2000c) reconoce las dificultades de implementar las RNs como sentencias de programas.

Una buena gestión pudiera ser permitir la facilidad de localizar el bloque de código correcto, pero las pruebas, integración, y todas las demás actividades que se requieren pueden llegar a ser un trabajo costoso, incluso en un pequeño cambio.

Problemas en la gestión de las reglas de negocio codificadas en el programa

Las RNs tienen un efecto significativo en la escalabilidad de la aplicación del negocio. Mientras se define el ambiente del negocio se encuentra mucha dificultad para gestionar y mantener las RNs si no están debidamente tratadas, especialmente si están escondidas en el código del programa, de esta manera los problemas más comunes que se derivan son (Bajec and Krisper, 2001):

- Cada cambio de las RNs requiere programación.
- Las RNs son distribuidas a través de una aplicación lógica; así el lugar donde hay que hacer el cambio es difícil de encontrar.
- Las RNs son un grupo lógico, dependientes e interrelacionadas. Por tanto ellas deben ser modificadas cuidadosamente considerando los posibles efectos en otras reglas.
- Los requerimientos para el cambio primario se alejan de las necesidades del negocio con las cuales los desarrolladores no están familiarizados, y corren el riesgo de que las RNs no sean entendidas e implementadas correctamente.
- Es muy difícil controlar las RNs si no existe un patrón único y común para ello.

Las RNs deben ser fáciles de acceder, ver, modificar y gestionar por ambas partes, desarrolladores y usuarios del negocio, de modo que resultarían mejores si existen aplicaciones que las hagan mucho más fáciles de modificar y mantener.

- Un factor importante de las RNs es que muchas son declarativas y no procedurales, describen el estado posible: requerido o prohibido. Pero no describen los pasos que se siguen para lograr o prohibir la transición de un estado a otro.

Frecuentemente existe un error muy común de clasificar requerimientos que no tengan nada que ver con el proceso del negocio como si fueran RNs.

3.1.2 Secuencia de Comandos (Scripts)

La esencia del *script* es la incrustación de un comportamiento variable en una estructura relativamente fija; no es sólo para las páginas Web, es una idea que se puede aplicar en cualquier lugar.

Lo importante de esta técnica es que se ubica el código de la regla fuera del código del programa. Si se quieren cambiar los límites a una regla, sólo se necesita cambiar el script. No se tiene que cambiar o probar, cualquier nuevo lanzamiento de los módulos de software más importantes.

La gran ventaja de usar scripts es la facilidad con que las decisiones basadas en reglas pueden ser separadas del código del programa. Si es posible trabajar en una situación de bajo rendimiento, los script pueden ser una buena opción.

El desempeño es el punto más débil de esta alternativa, pues cargar el conjunto de sentencias es una operación lenta, y aún más lenta, si el lenguaje es interpretado. La importancia que se le debe dar al desempeño depende de la situación en particular que se esté analizando.

3.1.3 Componentes

Un componente de regla está dedicado a la implementación de uno o más conjuntos de reglas específicas. Es una solución intermedia entre la línea de código y los motores de reglas, que son tratados después. El código dentro del componente de regla está escrito para satisfacer un conjunto limitado de objetivos, y no es la intención de volver a utilizarlo para otros conjuntos de reglas. La implementación interna está oculta a los clientes de la componente de regla, y la funcionalidad está disponible sólo a través de sus interfaces publicadas. Esto hace que sea fácil de cambiar la implementación, proporcionando que las interfaces originales se conserven.

Los componentes de reglas tienen dos ventajas sobre un motor de reglas de propósito general:

- El procesamiento interno se puede concretar a un problema específico, lo que permite un mayor rendimiento.
- Se evita la necesidad de hacer una inversión o un compromiso a cualquier motor de reglas particular.

Su desventaja radica en que la necesidad de reunir los datos pertinentes para comunicarse con el componente de regla aumenta el costo de una aplicación que utilice las reglas, obteniéndose un costo relativamente alto de invocación pero las interacciones locales son muy eficientes.

1.2.4 Mecanismos de bases de datos

Las bases de datos se presentan como repositorios de información central para los negocios y son sujetas a una amplia gama de políticas, reglas y requerimientos del propio negocio que son implementados dentro del sistema de base de datos, esto es reconocido en Atallah and Tompa 2011) que muestra un marco de trabajo de modelado de propósito general y de administración de restricciones que permite integrar numerosos aspectos del negocio dentro de la base de datos.

Las reglas centradas en datos probablemente se adaptarán más naturalmente dentro de la base de datos donde puedan tener un contacto más directo con los datos. El tipo de tecnología de base de datos más ampliamente utilizado en la actualidad es aquel que soporta el modelo relacional.

Sin embargo, esto no significa que siempre se puede exportar directamente la realización de una regla de un sistema de gestión de base de datos a otro. Si se desea implementar la regla de negocio en productos diferentes, será necesario, en la mayoría de los casos, implementarla de nuevo.

Todos los productos gestores de datos permiten almacenamiento y recuperación de información y mecanismos para controlar los datos. Es muy probable que en relación con las RNs se desee utilizar las reglas que giran en torno a la funcionalidad CRUD (crear, leer, actualizar y eliminar).

3.1.4.1 Reglas de negocio y restricciones de integridad en bases de datos

Una restricción de integridad es una expresión booleana que tiene que ver con algún requerimiento de una base de datos y debe ser evaluada siempre a *verdadero*. Tal restricción puede ser considerada como la expresión formal de alguna "regla de negocio" (Date, 2004).

Las RNs expresadas en lenguaje natural, también son a veces referidas como restricciones de integridad.

Las restricciones deben ser formalmente declaradas en el Sistema de gestión de bases de datos (SGBD), y el SGBD debe hacerlas cumplir, la declaración de estas se realiza usando el lenguaje de la base de datos, hacerlas cumplir es asunto del monitoreo del SGBD y si las restricciones son violadas las rechaza.

Para hacer cumplir esta restricción, el SGBD tendrá que supervisar todas las operaciones de inserción implicadas en una tabla o cambiar el estado existente.

Por supuesto, cuando una restricción es declarada, el sistema debe comprobar que la base de datos actualmente lo satisface, si no es así la restricción debe ser rechazada.

Cuando el SGBD asume una restricción la asunción de esta restricción es aceptada por él, será registrada en el catálogo bajo aquel nombre, y este nombre aparecerá entonces en mensajes, diagnósticos producidos por el sistema en respuesta a tentativas de violarlo.

Por supuesto, también se necesita un modo de deshacerse de restricciones existentes si estas no son necesarias, por ello existen sentencias para eliminar restricciones.

En resumen esta sección es una dificultad reconocida por: una restricción formalmente declarada es un predicado, cuando aquella restricción es comprobada, los argumentos son sustituidos por los parámetros y el predicado es así reducido a una proposición, y aquella proposición es entonces evaluada a verdadero.

Reconocen los tipos de restricciones: tipo, atributo, de tupla y de bases de datos.

Generalmente las RNs son expresadas con los tipos de restricciones: Restricciones de tupla, involucra solamente una tupla, y restricciones de base de datos que involucra varias relaciones.

Las restricciones en una tupla pueden ser restricciones de estado o restricciones de transición, que pueden ser entendidas como restricciones del estado de una tupla antes y después de la actualización.

La mayoría de los gestores de bases de datos cuentan con las restricciones Check y los disparadores para implementar las RNs de restricción.

3.1.4.2 Restricciones (Constraints) CHECK:

Las restricciones CHECK exigen la integridad del dominio mediante la limitación de los valores que puede aceptar una columna para cada tupla: dentro de un rango específico, de una lista de valores posibles, o conformado con un formato definido. También se pueden chequear las interrelaciones entre columnas de una misma tabla (Morgan, 2002d. Se puede crear una restricción CHECK con cualquier expresión

lógica que devuelva TRUE (verdadero) o FALSE (falso) basándose en operadores lógicos (Oppel and Sheldon, 2008).

Ventajas

La incorporación de las restricciones en la base de datos evitará cualquier acción que incumpla la regla: no sólo cuando el elemento se crea, sino también si se intenta actualizar en el futuro. Es posible aplicar varias restricciones CHECK a una sola columna y a varias columnas si se crea a nivel de la tabla. Así se pueden comprobar varias condiciones en un mismo sitio (MSND, 2008).

Desventajas

La principal desventaja de estas restricciones simples es que no se puede utilizar para condiciones más complicadas que implican varias tablas en un sistema de base de datos relacional.

3.1.4.3 Disparadores (triggers)

Las restricciones de integridad son a menudo implementadas proceduralmente, usando disparadores (*triggers* en inglés), los que son procedimientos pre compilados, que son almacenados en las bases de datos y son invocados automáticamente siempre que algún evento ocurra.

El evento puede ser cualquier instrucción UPDATE, INSERT o DELETE en una tabla o una vista.

A pesar de que el *disparador* sucede por un evento en una tabla específica, no se limita al control de las características de esa tabla solamente, sino que se puede utilizar para aplicar restricciones dentro de una tabla, a través de tablas, a través de bases de datos, e incluso a través de servidores. Amplían la lógica de comprobación de integridad, valores predeterminados y reglas del estándar SQL, aunque se deben utilizar las restricciones Check y los valores predeterminados siempre que estos aporten toda la funcionalidad necesaria (Melton and Simon, 2002). (Date, 2004)

1. En general, CREATE TRIGGER específica, entre otras cosas, un evento, una condición y una acción:
 - a. El evento es una operación sobre la base de datos, es decir sobre una tabla de la BD.
 - b. La condición es una expresión booleana que es evaluada a verdadero para que la acción sea ejecutada.
 - c. Las acciones están en el procedimiento que se dispara, el evento y la condición son a menudo considerados uno y llamado evento de disparo.

2. La combinación de evento, condición y acción se llama usualmente disparador. Los disparadores son generalmente llamados reglas evento-condición-acción, o ECA (del inglés *event-condition-action rules*).
3. Los posibles eventos incluyen INSERT, DELETE, UPDATE. Pueden incluir un atributo específico, se alcanza el final de la transacción (*COMMIT*), *alcanza un tiempo especificado* o se viola una restricción y no se aborta la transacción.
4. La acción puede ser realizada antes, después o en vez del evento especificado (AFTER, or INSTEAD OF).
5. En general, la acción puede ser realizada, para cada fila o para cada sentencia (FOR EACH ROW or FOR EACH STATEMENT).

Una base de datos que tiene asociados disparadores es llamada frecuentemente base de datos activa.

Ventajas

Pueden incluir instrucciones SQL complejas. También son útiles para exigir la integridad referencial, que conserva las relaciones definidas entre tablas (MSDN, 2008).

Desventajas

- Los factores desencadenantes son más lentos que las restricciones simples.
- También es necesario tener en cuenta el costo de la energía intelectual requerida para crearlos.

Además de las restricciones y los disparadores se reconocen los procedimientos y las vistas como recursos de bases de datos para implementar RNS (Morgan, 2002f).

1.2.5 Los procedimientos almacenados

Los procedimientos almacenados pueden facilitar en gran medida la administración de la base de datos y la visualización de información sobre esta y sus usuarios. Es una colección pre compilada de instrucciones SQL e instrucciones de control de flujo opcionales, almacenadas bajo un solo nombre y procesadas como una unidad; se guardan en una base de datos, permitiendo ser ejecutadas desde una aplicación.

Pueden aceptar parámetros y proporcionar sus resultados, devolver conjuntos individuales o múltiples y retornar valores (SQLServer 2000, 2004).

Ventajas

- Ejecución de una serie de instrucciones SQL en un único procedimiento almacenado.
- Referenciar a otros procedimientos almacenados desde el propio procedimiento almacenado, con lo que se pueden simplificar una serie de instrucciones complejas.

El procedimiento almacenado se compila en el servidor cuando se crea, por tanto, se ejecuta con mayor rapidez que las instrucciones SQL individuales. (MSDN, 2008).

1.2.6 Vistas

Una vista es una tabla virtual cuyo contenido está definido por una consulta. Al igual que una tabla real, una vista consta de un conjunto de columnas y filas de datos con un nombre. Suelen utilizarse para centrar, simplificar y personalizar la percepción de la base de datos para cada usuario (Melton and Simon, 2002).

Ventajas

Las vistas se pueden utilizar para realizar particiones de datos y para mejorar el rendimiento cuando estos se copian. Además, permiten a los usuarios centrarse en datos de su interés y en tareas específicas de las que son responsables. Los datos innecesarios pueden quedar fuera de la vista; de ese modo, también es mayor su seguridad, puesto que los usuarios solo pueden ver los definidos en la vista y no los que hay en la tabla subyacente (MSDN, 2008).

Los motores de reglas de negocio

Se habla de motores de RNs como aplicaciones de software que contienen definiciones de RNs y las ejecutan.

Según el propio Ronald Ross, existe variación entre motores de RNs, y los ha clasificado como aquellos que principalmente tratan con bases de datos, y aquellos que son usados para hacer inferencias. (Ross, 2003).

La mayoría de los SI tienen como principal componente los datos y la importancia de estos recae en que ellos reflejan en gran medida la lógica que el negocio requiere debido a que los hechos almacenados reflejan las RNs y son sensibles a transacciones comerciales.

Sin embargo algunos motores de reglas son usados principalmente para hacer inferencias para proporcionar al usuario la información que este requiere conocer, ellos son muy comercializados en la actualidad (Chisholm, 2006). El motor de reglas puede generar una respuesta a la pregunta del usuario. Generalmente cuando en la

literatura se hace referencia a motores de reglas se hace referencia a estos últimos. Sin embargo, el propio Chisholm (2006) reconoce, la utilidad de conformar motores de reglas de bases de datos.

1.2.7 Los motores de reglas de inferencia

Un motor de reglas, o servidor de la lógica, es un componente diseñado con el único propósito de gestionar reglas. No se construye para resolver cualquier problema particular sino que brinda un conjunto genérico de capacidades para definir, almacenar y aplicar reglas.

Se construyen generalmente en la forma de un servidor lógico, con un lenguaje de RNs y un estilo de operación.

Estos servidores, además de manejar la lógica del negocio proporcionan una serie de funciones útiles para el trabajo con las aplicaciones. Las funciones típicas son:

- Características de elasticidad para aumentar la disponibilidad del sistema.
- Almacenamiento en caché de datos para reducir los tiempos de respuesta.
- Las características adicionales de seguridad para proteger los sistemas centrales en un entorno global de usuarios.
- Diversas técnicas, tales como el uso común de recursos, para aumentar la escalabilidad de la aplicación.
- Conectores listos para interactuar con orígenes de datos específicos.
- Funciones transaccionales y de bloqueo de registros.

La Figura 1 muestra los elementos principales de un motor de reglas de propósito general. El origen de las reglas se define usando un lenguaje de reglas adecuado y, por lo general, se compila en una representación interna eficiente. Las reglas compiladas se almacenan en una estructura de datos propia, disponibles para su uso por un mecanismo de inferencia, que comprende la estructura y es capaz de aplicar las reglas en tiempo de ejecución.

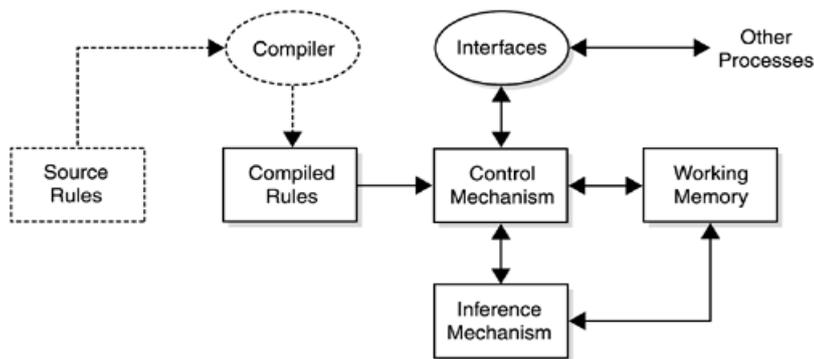


Fig 1. Elementos principales de un típico motor de reglas (Morgan, 2002f)

Algunos motores de reglas proporcionan herramientas adecuadas para permitir a los desarrolladores dar un vistazo a lo que está pasando, principalmente para propósitos de depuración, pero por lo general no tienen relación directa con el mundo exterior. El papel principal es desempeñado por un mecanismo de control encargado de coordinar las acciones. El uso del motor de reglas ya incluye un lenguaje que se usa para expresar las mismas y generalmente una interfaz interactiva para su construcción.

Como las reglas revelan y restringen las políticas comerciales y la tecnología de RNs proporciona un medio automatizado para este comportamiento comercial, entonces el uso de BRMS (del inglés *Business rules management system*) para un manejo centralizado de las RNs y los servidores para las aplicaciones en que necesitan las reglas (a través de la empresa) ofrece un papel principal para la integración de aplicaciones. (Nelson *et al.*, 2009)

Expresiones de las reglas

La mayor diferencia entre los motores de reglas es la forma en que expresan las reglas, el enfoque principal utiliza un lenguaje especializado, estos lenguajes se caracterizan porque: las declaraciones de reglas son declarativas y no procedurales, definen las relaciones entre los hechos, no la forma de hacer algo; no hay un control de flujo predeterminado; las reglas se activan según el estado actual del problema.

El uso del motor de reglas ya incluye el lenguaje que se usa para expresar las reglas y generalmente una interfaz interactiva para su construcción.

Se usa con frecuencia una interfaz de programación de aplicaciones (API) que permite que el código externo participe en diversos tipos de interacciones con el motor de reglas. En algunos casos, el motor de reglas también controla la interfaz de usuario.

3.1.8.1 Mecanismos de inferencia

Las reglas pueden agruparse en conjuntos de reglas para centrarse en la búsqueda de una solución a un problema específico. Los motores de reglas proporcionan una forma de trabajo, donde pueden compartir un ambiente común y un medio común de expresión. También comparten un mecanismo de inferencia común, ya sea guiado por datos o por objetivos.

Despliegue.

Existen varias opciones de tecnologías para la construcción de un motor de reglas dentro de un sistema operacional, se incluyen estrategias ingeniosas como la clonación hacia varios servidores así como brindar el motor de reglas como un conjunto de funciones pertenecientes a una biblioteca (Morgan, 2002b).

En este sentido se han realizado estudios recientes basados en comparar los niveles de despliegue de diferentes BRMS, con estándares de integración de sistemas usando el modelo de madurez de integración de aplicaciones (EAI). (Nelson *et al.*, 2009)

Ventajas

Con los motores de reglas se pueden ahorrar esfuerzos en la implementación, permitiendo manejar conjuntos de reglas grandes y complejas. Es probable obtener un repositorio centralizado: permitiendo cuidar de sus reglas en una forma organizada.

Desventajas

La adopción de un motor de reglas requiere una inversión significativa aunque se cuente con tiempo suficiente.

El uso exclusivo del motor para trabajar con RNs, incluye la actitud de algunos analistas que adoptan la posición de que sólo las reglas del motor de reglas son las adecuadas y todas las demás pueden ser ignoradas, u obliga a todas las reglas a implementarse en el motor de inferencia. Esto sería un gran error (Morgan, 2002e), pues algunas reglas se pueden implementar de manera más simple o mejor fuera de este.

3.1.9 Sistemas de flujos de trabajo

Uno de los elementos más importantes de un modelo de negocio es la definición de los procesos de negocios que funcionan dentro de la organización. La automatización de procesos de negocio se encuentra bajo el nombre de flujo de trabajo.

Las herramientas de flujo de trabajo generalmente ofrecen los tipos de funciones siguientes:

- Edición interactiva de definiciones de procesos, utilizando una presentación gráfica de un diagrama de flujo.

- Crear instancias de procesos, definidos en una estructura de control que puede controlar la actividad relevante casi en tiempo real.
- El seguimiento de las actividades de los distintos participantes en el proceso.
- La información de ruta entre los participantes y otros recursos, incluso haciendo cola en los puntos apropiados donde puede haber limitaciones de recursos.
- Reconociendo los acontecimientos externos, y los avances de las acciones internas, a fin de mantener una visión precisa del estado del proceso.
- La invocación de las unidades externas de software para introducir sus capacidades en los puntos apropiados.
- Proporcionar informes y pantallas para apoyar la gestión y la supervisión de los procesos de negocios.

Las reglas tienen un papel importante que desempeñar en la definición y el control del flujo de trabajo, asociadas típicamente con los puntos de ramificación en el proceso, donde se pueden tomar caminos alternativos, dependiendo de las condiciones que se aplican. Las condiciones en las ramas son llamadas *guardianes*, o *porteros*, debido a su papel en el control de la entrada y condiciones de salida para las diversas etapas del proceso (Morgan, 2002b). En un contexto de flujo de trabajo, las reglas están más fuertemente asociadas con el control de la conducta.

Las reglas que se aplican internamente tienden a estar en un formato propietario que es parte del producto de flujo de trabajo.

Desventajas

Es bastante útil para implementar casos sencillos de reglas pero se ve bastante limitado para enfrentar casos complejos, donde debe depender del empleo de otro de los mecanismos para su solución.

1.2.10 Tablas de búsqueda

Estas tablas contienen conocimiento pre compilado en una forma que sea rápido y conveniente de acceder.

Una tabla de búsqueda se puede utilizar para expresar la lógica de negocio en una forma que es más fácil de usar que las reglas originales. Las reglas de la forma *if.. then* pueden representarse de manera que las implicaciones de una o más RNs están codificadas en una tabla para que la aplicación pueda simplemente leer el resultado correspondiente. Se necesita definir qué tipo de evento, en qué proceso afecta al negocio y si produce un impacto.

Por ejemplo, decidir qué procesos podrían verse afectados por un evento es sólo una cuestión de recuperar todas las filas con el nombre del evento relevante y un sí en los impactos.

Aunque las tablas pueden ser convenientes, su uso excesivo puede hacer que el sistema sea difícil de mantener.

Un ejemplo de tabla de búsqueda podría ser el siguiente:

Se tienen un conjunto de reglas que determinan si un evento influye o no en un proceso. Como resultado de codificar las reglas se tiene la tabla siguiente.

Tabla 2. Tabla de Eventos y procesos

Evento	Proceso	Impacto
E1	P1	Si
E1	P2	No
E1	P3	Si
E2	P1	Si
...

El número de filas y columnas de esta tabla puede reducirse considerablemente adoptando variantes como: incluir las filas en las que el impacto es positivo.

Existen diferentes alternativas en cuanto al almacenamiento de la tabla, el almacenamiento en una tabla de base de datos relacional, o el almacenamiento en memoria.

En ambos casos es necesario definir un mecanismo para generar la tabla (Morgan, 2002f).

Desventajas

Dada la posibilidad de producir tablas sin reflejar las sentencias de reglas, con el tiempo pueden llegar a ser muy difíciles de mantener. Si hay cambio de analista durante el desarrollo del trabajo puede ser muy perjudicial ya que en muchos casos las entradas no son absolutamente evidentes.

1.3 Tendencias en la gestión de reglas de negocio

Se ha demostrado que la técnica de adicionar parámetros a la aplicación y sus componentes es aplicable al manejo de las RNs para adicionar flexibilidad y adaptabilidad a una aplicación. Estos parámetros pueden darse en un conjunto de ficheros o en una BD y pueden ser administrados a través de un utilitario de

configuración. De modo que la aplicación pueda ser adaptada a diferentes ambientes y situaciones sin ningún esfuerzo de programación. Aunque las políticas del negocio se quedan escondidas en la lógica de la aplicación, pueden ser modificadas a través de parámetros sin ninguna necesidad de cambiar el código del programa.

Pero, hay muchos aspectos a tener en cuenta en la implementación de las reglas que están relacionados con los abordados por Bajec and Krisper:

¿Cómo aplicar RNs?. ¿Qué tecnología usar? ¿Cuál es el mejor lugar para la aplicación de RNs (para minimizar el esfuerzo necesario para cambiar las RNs)?

La gestión de RNs incluye: ¿dónde almacenar las RNs (incluyendo su contexto)? ¿Cómo hacer para mantener los cambios de RNs y las versiones? ¿Cómo gestionar las RNs para toda una organización?.

En respuesta a estas interrogantes se han presentado tendencias de desarrollo de herramientas para manejar las RNs, (Bajec *et al.*, 2000a) teniendo en cuenta cómo se representan, implementan y ejecutan las RNs, estas herramientas se pueden clasificar como:

- Herramientas independientes de bases de datos: las RNs son implementadas en una BD, usando disparadores y procedimientos almacenados. Sin embargo estos son generados automáticamente y manejados por herramientas que no son gestores de bases de datos.
- Herramientas basadas en el servidor: las RNs son creadas por herramientas de desarrollo que se convierten en una capa intermedia de servicios de aplicación y residen sobre un servicio de aplicación.
- Sistemas basados en reglas: En vez de especificar restricciones sobre elementos de datos o tabla, un método orientado a la lógica captura la lógica del negocio a un alto nivel y las reglas asociadas. En tiempo de ejecución son usados motores especiales para procesar las reglas y generar respuestas apropiadas.

1.3.5 Justificación para implementar las de reglas de negocios en bases de datos

Las soluciones para implementar RNs en bases de datos han sido tratadas por algunos autores, aceptándolas como una manera de evitar ubicarlas de forma estática en el código de la aplicación debido a que (Bain, 2009):

1. Las reglas evolucionan.
2. Las situaciones cambian y hay nuevas regulaciones.
3. Las reglas pueden ser usadas por más de una aplicación.

Así los programadores de las aplicaciones no son responsables de llevar el control cada vez que una regla sea añadida o actualizada.

Si se trabaja con una aplicación de dos capas, se puede tener la base de datos y las funciones almacenadas para ejecutar las reglas de negocio implementadas o creadas dentro del propio gestor, de modo que:

1. Las RNs están disponibles para cualquier negocio o programa que use esta base de datos.
2. Las RNs pueden ser actualizadas o adicionadas sin tener que cambiar la aplicación.
3. El programador tendrá un conjunto de reglas potentes y adaptables que son independientes de la aplicación.

También pueden lograrse efectos similares con una aplicación de tres capas, donde las RNs también se pueden implementar en el propio gestor.

3.2.1.1 Operaciones CRUD y reglas de negocios

Las operaciones CRUD para crear, recuperar, actualizar y eliminar, se usan para todos los sistemas, sin embargo las RNs nunca deben expresarse en términos de estas operaciones, ya que con esta terminología siempre la referencia se hará directamente a una operación de sistema o un dato. Una regla formulada en estos términos limita la cobertura del evento donde la regla de negocio debe ser evaluada, (Ross, 2010).

La regla no debe ser expresada como el ejemplo siguiente:

Regla de actualización: Actualice el costo del producto si el costo de cualquier componente cambia.

Aquí se está limitando el punto de activación de la regla al hecho del cambio de los componentes, sin embargo desde el negocio esta regla debe ser expresada como:

El costo de un producto debe ser calculado como la suma del costo de todos los componentes del producto. Por tanto no cubre solamente el cambio sino también:

- Cuando un componente es recién añadido a la composición de un producto.
- Cuando un componente está recién retirado de la composición de un producto.

Cuando una regla se formula en función de una operación CRUD se está imponiendo una condición técnica de cuándo se debe evaluar, sin embargo la experiencia ha demostrado como la gran mayoría de las RNs inherentemente se disparan en varios puntos o infieren automáticamente cuando el código asociado a la regla debe ser evaluado, de modo que descubre los eventos donde las reglas deben ser evaluadas.

1.3.2 Consideraciones para la implementación automática de reglas de negocios

Una de las actividades más importantes en el desarrollo de SI basados en reglas, es la consideración de las diferentes tecnologías y herramientas disponibles para apoyar la implementación de reglas, incluida la generación de código (Bajec *et al.*, 2000b). Estas herramientas van desde las orientadas a bases de datos, pues hacen cumplir las reglas utilizando los mecanismos de bases de datos, como disparadores y procedimientos almacenados, vistas para hacerse cargo de la ejecución de la regla, hasta los motores y máquinas de reglas más sofisticados.

Algunas de las cuestiones que continúan siendo desafíos para la implementación de reglas han sido analizadas (Bajec and Krisper):

¿Qué tecnología se debe utilizar? lo que depende de varios factores, pero particularmente del tipo de sistema en desarrollo.

Por ejemplo, en una aplicación típica basada en la aplicación del conocimiento, las reglas se capturan y se almacenan en una base de reglas y se ejecutan por un motor de reglas.

Sin embargo en un sistema típico de flujo de trabajo, las RNs se integrarán en la definición del flujo de trabajo, y será utilizado un motor de flujo de trabajo para ejecutar el mismo.

Otro de los desafíos en la implementación de RNs es proporcionar una herramienta de apoyo para la generación de código de programa (para la ejecución de las RNs) a partir de las especificaciones de RNs.

Las herramientas más comercializadas que están disponibles hoy día y ofrecen este tipo de funcionalidad son para determinadas tecnologías. Por ejemplo, si se utiliza una herramienta que utiliza un motor de reglas de proceso, el código de programa sólo puede ser generado para el uso en ese motor de reglas. Además, dado que las especificaciones deben ser rigurosas a interpretarlas por una computadora, las herramientas a menudo sólo admiten un número limitado de categorías de reglas.

Con Athenas (Zimbrão *et al.*, 2003) presenta un motor de ejecución de RNs sobre el sistema de gestión de Oracle, para implementar automáticamente las reglas que restringen el comportamiento del negocio, reconocidas como reglas de restricción.

1.4 Repositorio de reglas

Aunque las reglas no se identifiquen explícitamente ellas existen, pero probablemente están escondidas dentro de la aplicación de modo que se hace muy difícil de localizar y mantener, en los casos de los motores de reglas están almacenadas dentro del motor de reglas, entendibles solo por este y poco adaptable.

Una de los componentes fundamentales para trabajar con RNs para lograr algún grado de independencia de las reglas de los SI es el repositorio de reglas. El repositorio almacena la declaración de las reglas y aspectos que permiten garantizar su control y mantenimiento. Existen en la literatura numerosos esquemas de repositorios de reglas. Morgan (2002e) hace una propuesta que se corresponde con la de un repositorio de reglas para todas las reglas independientemente de la implementación de las mismas. Se propone tener el control de todas las reglas con las que se trabaja y se logra relacionar con la implementación de cada una de ellas. La descripción de esta propuesta se puede ver en el Anexo 1.

Cada RN debe describirse en el repositorio por un conjunto de atributos, entre los más importantes se deben describir: el identificador de la regla, número de la versión de la regla correspondiente, la fecha de creación o del último cambio, el tipo de regla, el estado, la sentencia de la regla en sí. Se destaca que las reglas pueden ser aplicadas por determinado período de tiempo.

Una característica de esta propuesta es que el repositorio muestre trazabilidad desde los modelos de negocio que soportan su definición primaria a sus realizaciones en el sistema de información.

1.5 Influencia de las modificaciones de las reglas en los SI

Los cambios en el entorno empresarial de una organización casi siempre pueden ser impulsados tanto desde las decisiones internas de la gestión de la organización o de fuerzas externas, como las leyes y regulaciones del gobierno. Estos cambios con frecuencia requieren reflejarse en los sistemas nuevos o modificados (Bajec *et al.*, 2005b).

Generalmente los cambios en los procesos de negocio y en los sistemas de apoyo se reflejan en los cambios de las RNs y sus implementaciones, que son revisadas y modificadas de acuerdo a los nuevos objetivos, metas y políticas. Estos cambios deben ser coordinados a nivel de empresa. Una RNs en particular puede estar involucrada en los procesos de negocio de varios sistemas y apoyada por un conjunto de subsistemas. Con el fin de ser capaz de mantener el apoyo a los sistemas en consonancia con los requerimientos del negocio, debe tenerse documentación de

cómo las RNs evolucionan a partir de su origen en el entorno empresarial para su aplicación en el SI (Bajec *et al.*, 2005b).

En el análisis de las RNs tipo restricción, cuando estas se modifican pueden tornarse en otra similar o en una regla totalmente diferente; para garantizar la consistencia del negocio y en correspondencia del SI asociado, se debe controlar el conjunto de acciones permitidas para estas modificaciones.

Cuando las reglas han sido implementadas junto a los datos, estas acciones pueden involucrar recursos como disparadores y vistas. Los datos afectados por la transformación de una regla, que llega a ser una nueva, pueden ser tratados en dependencia de los intereses de los usuarios del negocio: algunos datos pueden necesitar cumplir la regla anterior, otros acatar la nueva regla producto de la transformación resultante.

3.4.1 Control de cambios y versiones de reglas de negocio

Según Bajec (Bajec *et al.*, 2005a), además de las actividades que se realizan durante el modelado de la empresa y el desarrollo de un SI, el escenario manejador de RNs prescribe tareas adicionales que se ocupan de los cambios de RNs a través de su ciclo de vida. Estas actividades se pueden realizar a nivel de empresa o del SI. Ellos incluyen:

- **Control de cambios:** El propósito de esta actividad es coordinar los cambios de las RNs. En general, el motivo de cambio en las RNs siempre se plantea desde el entorno de negocio de la empresa. Si parece que una RN ha cambiado debido a algún problema técnico o a causa de algún nuevo requisito del SI, y desde el punto de vista de negocio no es necesario el cambio, entonces esto no es realmente una RNs. Las RNs son propiedad de los negocios y siempre están estrechamente relacionadas con el entorno empresarial. Se recomienda que, por cada cambio en el modelo de RNs debe haber una explicación a nivel empresarial, describiendo por qué el cambio es necesario. Para ser capaz de controlar los cambios, la información tiene que ser dirigida sobre quién ha solicitado los cambios, quién los ha aprobado y cuándo serán implementados.
- **El control de versiones:** Debido a su naturaleza dinámica, las RNs pueden tener varias versiones sobre el tiempo. En algunos casos, incluso pueden existir varias versiones de la misma RNs en uso del SI de la empresa. Por ejemplo, una versión de la regla puede ser utilizada en un subsistema, mientras que otros subsistemas están utilizando una versión diferente. A fin de conocer las versiones que se utilizan en un determinado sistema, este último debe ser

capaz de realizar evaluaciones de diferentes versiones de la regla y darle seguimiento a la historia de las RNs.

- Control de impacto: Las RNs rara vez son independientes, lo que significa que un cambio en una regla en particular puede causar cambios en otras reglas. Para gestionar los cambios, todas las dependencias entre las reglas y otros componentes se debe conocer las interdependencias, o sea darle seguimiento. Antes que una RNs sea cambiada, se debe hacer un análisis de impacto para averiguar si existe algún obstáculo en el cambio de la regla.

1.5.2 Versiones e historial de reglas de negocio

Una *versión* es una *instantánea significativa* de un objeto de diseño creado en un momento del tiempo. Aparte de la semántica atribuida al concepto de versión, una versión de un objeto dado se crea a partir de las modificaciones hechas en el tiempo a las versiones previas, partiendo de una versión inicial (García Pérez, 1997). El hacer versiones, aún en su forma más simple, requiere que se mantenga una historia de los cambios para permitir la retención de la definición de reglas pasadas. Según García Pérez (1997) la historia de versiones puede ser lineal (en forma de *backup*) pero un versionamiento más poderoso debe suministrar la posibilidad de manejar versiones alternativas, y por esto la historia de versiones se convierte en un árbol. El versionamiento incluye que no se pierda información valiosa, por ejemplo, si cambia la restricción para un atributo. Para ello es preciso definir reglas de integridad y mecanismos que controlen la evolución.

Las investigaciones sobre las versiones de reglas, en el enfoque de RNs, están basadas en la evolución constante a que pueden estar sujetos los negocios. Del estudio realizado sobre este tema la Dra. Silvie Spreeuwenberg tiene experiencia con la modelación de RNs, lo cual ha sustentado el desarrollo de herramientas y técnicas para aumentar la calidad de las mismas. Silvie expresa, "... *uno debe concentrarse en la calidad de la gestión de RNs para lograr el propósito del enfoque de reglas de negocio*" (Spreeuwenberg, 2008; Spreeuwenberg, 2007b; Spreeuwenberg, 2007a) . En (Spreeuwenberg, 2007a, Spreeuwenberg, 2007b, Spreeuwenberg, 2008) se propone resolver el problema de versiones de reglas de una manera declarativa, partiendo de que todas las reglas están sujetas a ser versionadas y como condición extra se tiene que pueden estar activas en un período de tiempo determinado, se utiliza un atributo de marca de tiempo (*markdate*) con la fecha de comienzo y fecha fin en que la regla estuvo activa.

En Spreeuwenber (2007a) se muestran los enunciados siguientes que describen el problema de versiones de reglas:

- Las reglas son sentencias declarativas que son usadas en una o más tareas para determinar un cierto valor para un atributo.
- Algunas reglas solo pueden ser aplicadas en un período de tiempo determinado. Este período puede ser descrito con fecha de inicio y fecha fin. La fecha fin no es obligatoria. Las reglas que solo tienen fecha de inicio son aplicables por lo menos hasta la fecha actual.
- Las reglas aplicables sobre cierta fecha deben ser consistentes.
- Un caso del pasado puede ser retomado aplicando las reglas que estaban activas en ese período.
- La fecha que se usa para recuperar las reglas correctas pueden ser diferentes de acuerdo con: La tarea que se va a llevar a cabo, el caso (situación) sobre el cual las reglas se aplicarán, o el evento que produjo la activación de las reglas. Spreeuwenberg (2007b) en aras de la gestión de las reglas las agrupa en el conjunto *RulesetPostRules*, que pueden ser activadas en el mismo período de tiempo. Se utilizan algoritmos de encadenamiento hacia adelante y hacia atrás; relacionados con los motores de reglas que tienen un mecanismo de inferencia. La desventaja de esta solución está en el mantenimiento del conjunto de regla, el cual contendrá una buena cantidad de redundancia y reglas cuando hay poca superposición en el período de aplicabilidad de las mismas.

En (Spreeuwenberg, 2008) se discute una estrategia para tratar con versiones de reglas, la cual trabaja si hay lotes de diferentes versiones de nuevas reglas. Para versionar reglas complejas recomienda ver las reglas como un objeto con atributos (meta información) y crear un objeto para cada regla. Esto permite especificar una información extra acerca de la regla en su modelo de objeto.

Entre los principales atributos para las versiones de reglas se especifican:

- *StartDate* fecha a partir de la cual la regla es aplicable.
- *EndDate* fecha hasta la cual la regla es aplicable.
- *PostRule* atributo booleano que indica si la regla está activa o no.
- *RuleStatement* sentencia o referencia de la regla actual.

4. CONCLUSIONES

Los investigadores del mundo de las RNs tienden a categorizar las reglas para facilitar el conocimiento de estas y un tratamiento adecuado y efectivo de las mismas desde su origen, en los expertos del negocio, hasta las implementaciones en los SI.

Se observa que existen categorías de RNs involucradas con los datos del negocio, los cuales se almacenan generalmente en bases de datos relacionales. Estas reglas, pudieran ser en última instancia comprobadas o ejecutadas ante operaciones sobre los datos.

Con la tendencia de gestionar las RNs con diferentes tipos de herramientas que permiten su generación automática, se vislumbra la posibilidad de no forzar la implementación de todas las reglas en motores de reglas de inferencia.

Se evidencia una diversidad de lenguajes para formalizar la escritura de las RNs, cada uno de estos responden a un nivel de abstracción que puede corresponder a los niveles: informal o natural estructurado, técnico y formal (ya cerca de la implementación), algunos buscan la interoperabilidad de las reglas en la Web, se desarrollan lenguajes que responden al trabajo con motores de reglas específicos; sin embargo el costo de aprenderlos y usarlos es alto. Ninguno de los lenguajes estudiados puede aplicarse a escribir reglas en función de los términos del negocio que se correponden con las tablas y atributos de las bases de datos relacionales.

Al finalizar este estudio se puede considerar que adoptar uno de los motores de reglas de inferencia comercializados existentes para implementar automáticamente las reglas requiere una gran inversión de tiempo, y obliga en algunos casos a complejizar la implementación de reglas que pudieran implementarse de manera más sencilla fuera de estos, por ejemplo en forma de recursos de bases de datos activas.

Para la generación de las implementaciones de reglas de manera automática se deben tener en cuenta estrategias para controlar las posibles modificaciones de las reglas como el control de versiones y el historial de reglas.

5. REFERENCIAS

- ANDREESCU, A. & MIRCEA, M. (2009) Managing Knowledge as Business Rules. *Informatica Economică. Academy of Economic Studies, Bucharest, Romania*, 13, 63-74.
- ASHWELL, R. (2006) Define Business Rules. CRaG Systems.
- ATAULLAH, A. & TOMPA, F. W. (2011) Business Policy Modeling and Enforcement in Databases. *unpublished, www.cs.uwaterloo.ca/~aataulla/unpublished/BPM_in_Databases.pdf.*
- BAIN, M. A. (2009) How to Use Business Rules in a MySQL Database Using MySQL Functions to Create Two-Tier Business Savvy Applications
- BAISLEY, D. (2008) SBVR: What Are the Possibilities? *Business Rules Journal*, 9.
- BAJEC, M. & KRISPER, M. ISSUES AND CHALLENGES IN BUSINESS RULE-BASED INFORMATION SYSTEMS DEVELOPMENT.
- BAJEC, M. & KRISPER, M. (2001) Managing business rules in enterprises. *Faculty of Computer and Information Science*, Ljubljana, University of Ljubljana.
- BAJEC, M. & KRISPER, M. (2005) A methodology and tool support for managing business rules in organisations. *Information Systems*, 30, 423-443.
- BAJEC, M., KRISPER, M. & RUPNIK, R. (2000a) Using Business Rules Technologies To Bridge The Gap Between Business and Business Applications. IN RECHNU, G. E. (Ed.) *Proc. of the IFIP 16th World Computer Congress 2000, Information Technology for Business Management*. Peking, China.
- BAJEC, M., KRISPER, M. & RUPNIK, R. (2005a) A methodology and tool support for managing business rules in organisations. *Information Systems, Sep 2005*, 30, no. 6, 423-443.
- BAJEC, M., RUPNIK, R. & KRISPER, M. (2000b) Using Business Rules Technologies to Bridge the Gap Between Business and Business Applications. IN G. RECHNU BEIJING, C. (Ed.) *Proceedings of the IFIP 16th World Computer Congress 2000. Information Technology for Business Management*. Beijing, China.
- BAJEC, M., RUPNIK, R. & KRISPER, M. (2000c) USING BUSINESS RULES TECHNOLOGIES TO BRIDGE THE GAP BETWEEN BUSINESS AND BUSINESS APPLICATIONS. IN RECHNU, G. (Ed.) *Proceedings of the IFIP 16th World Computer Congress 2000. Information Technology for Business Management*. Beijing, China.
- BAJEC, M., RUPNIK, R. & KRISPER, M. (2005b) A methodology and tool support for managing business rules in organisations. *Information Systems*, 30, 423-443.
- BESEMBEL, I. M. & CHACÓN, E. (2001) Objetos y reglas de negocios en la integración y automatización de procesos de producción continua.
- BOLEY, H., PASCHKE, A. & SHAFIQ, O. RuleML 1.0: the overarching specification of web rules. *Lecture Notes in Computer Science*, 6403, 162-178.
- BRS (2004) Business Rules Solutions. BRS RuleSpeak® Practitioner's Kit. *Business Rule Solutions, LLC*.
- BRUEGGE, B. & DUTOIT, A. H. (2009) *Object Oriented Software Engineering Using UML, Patterns, and Java*, Prentice Hall.
- BUSINESS RULES GROUP (2003) Manifiesto de Reglas de Negocio. IN ROSS, R. G. (Ed.). Business Rules Group.
- CARVER, A. & MORGAN, T. Characterizing Business Rules for Practical Information Systems. *Enterprise, Business-Process and Information Systems Modeling*, 443-452.

- CARVER, A. & MORGAN, T. (2012) A Framework for Relating Business Constraints to Information Systems. *Enterprise, Business-Process and Information Systems Modeling*, 300-314.
- CHISHOLM, M. (Ed.) (2006) *How to Build a Business Rules Engine*, San Francisco, United States of America, Morgan Kaufmann.
- DATE, C. J. (2000a) *What Not How: The Business Rules Approach To Application Development.*, Addison Wesley Longman.
- DATE, C. J. (2000b) *What Not How: The Business Rules Approach To Application Development.*, Addison Wesley Longman Inc.
- DATE, C. J. (2001) Constraints & Predicates: A Brief Tutorial (Part 2). *Business Rules Journal*, V2.
- DATE, C. J. (2004) An introduction to Database Systems.Eight Edition. Eight (Octava) ed. United State, Pearson Education.
- DEMUTH, B. (2004) The Dresden OCL Toolkit and its Role in Information Systems Development. *13th International Conference on Information Systems Development: Methods and Tools, Theory and Practice*. Vilnius, Lithuania.
- DEMUTH, B. (2005) The Dresden OCL Toolkit and the Business Rules Approach. Technische Universität Dresden.
- GARCÍA GONZÁLEZ, C. E. (2009) Ayuda a la modelación conceptual y lógica de bases de datos relacionales. *Departamento de ciencia de la computación*. Santa Clara, Universidad Central "Marta Abreu" de Las Villas.
- GARCÍA PÉREZ, A. M. (1997) Un modelo de versiones para la construcción de software de ayuda al diseño. *Departamento de Ciencias de la Computación*. Santa Clara, Universidad Central "Marta Abreu" de Las Villas.
- GOTTESDIENER, E. (1997a) Business rules Show Power *Promise Application development trends*. EBG Consulting ed.
- GOTTESDIENER, E. (1997b) Business Rules Show Power, *Promise Application Development Trends*,. vol. 4., no. 3. EBG Consulting, Inc. , 36-42.
- GROUP, B. R. (2008) Business Rule Group, a non-commercial peer group of IT professionals. [Online]. .
- GROUP, B. R. (July, 2000) Defining Business Rules – What Are They Really? rev. 1.3, .
- HALPIN, T. (2006) Verbalizing Business Rules (part 16). *Business Rules Journal*, Vol. 7.
- HAY, D., HEALY, K. A. & HALL, J. (2000) Defining businessrules- what are they really? Technical Report 1.3. *The Business Rules Group*.
- HEIDENREICH, F., WENDE, C. & DEMUTH, B. (2008) A Framework for Generating Query Language Code from OCL Invariants. 9, 4-10.
- INVERNIZZI, H. OCL - Estado da Arte .Banco Central do Brasil. Ricardo Santos - Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa.
- LAGE, O. SWRL.
- LOWENTHAL, B. (2005) Rule Enabling Applications with Oracle Business Rules. Oracle Corporation.
- MARTÍNEZ HERNANDEZ, J. L. (2010) Introduciendo semántica en un proceso de desarrollo software a través de reglas de negocio. *Escuela técnica superior de ingenieros de telecomunicación* Madrid, Universidad politécnica de Madrid.
- MATEI, I. (2006) Implementing Business Rules with Software Agents. *Department of Computer Sciences*. Tampere, Finlandia, Universidad de Tampere.
- MELTON, J. & SIMON, A. R. (2002) *SQL1999: understanding relational language components*, Morgab Kaufmann.
- MORGAN, T. (2002a) Business Rules and Information Systems: Aligning IT with Business Goals. Addison Wesley.

- MORGAN, T. (2002b) *Business Rules and Information Systems: Aligning IT with Business Goals*, Addison Wesley.
- MORGAN, T. (2002c) *Business Rules and Information Systems: Aligning IT with Business Goals*, Indianapolis, USA, Addison Wesley.
- MORGAN, T. (2002d) *Business Rules and Information Systems: Aligning IT with Business Goals*. Indianapolis, USA, Addison Wesley.
- MORGAN, T. (2002e) *Business Rules and Information Systems: Aligning IT with Business Goals: Managing Business Rules and Models*. Capítulo 8., Addison Wesley, .
- MORGAN, T. (2002f) *Business Rules and Information Systems: Aligning IT with Business Goals: The technology environment* . Capítulo 7. Realizing Business Rules., Addison Wesley, .
- MSDN (2008) MSDN LIBRARY VisualStudio 2008. Microsoft Corporation.
- NELSON, M. L., PETERSON, J., RARIDEN, R. L. & SEN, R. (2009) Transitioning to a business rule management service model: Case studies from the property and casualty insurance industry.
- OMG (2008a) *Semantics of Business Vocabulary and Business Rules*.
- OMG (2008b) *Semantics of Business Vocabulary and Business Rules (SBVR)*.
- OMG (2010) *Object Constraint Language*. Object Management Group, Inc.
- OPPEL, A. & SHELDON, R. (2008) *SQL: a beginner's guide*, McGraw-Hill Profesional.
- PEREZ ALONSO, A. (2010) *Reglas de Negocio en Bases de Datos Relacionales. Bases de Datos*. Santa Clara, Universidad Central "Marta Abreu" de Las Villas.
- ROSS, R. G. (1997) *The Business Rule Book: Classifying, Defining and Modeling Rules, Version 4.0* Business Rule Solutions.
- ROSS, R. G. (2003) *Principles of the Business Rule Approach*, Addison-Wesley Professional.
- ROSS, R. G. (2010) What Is a Business Rule? *Business Rules Journal*, 11 1-4.
- ROSS, R. G. (Feb.2010) CRUD in Business Rules: Accident-Prone Decision Logic. *Business Rules Journal*, 11.
- SANTOS, R. & INVERNIZZI, H. (2004) OCL – Estado del Arte. . Facultad de Ciencias y Tecnología, Universidad Nova de Lisboa.
- SCOTT W., A. B. R. (2011) *Business Rules*.
- SPREEUWENBERG, S. (2007a) Rule History and Versioning (Part 1). *Business Rules Journal*, Vol. 8, No. 11
- SPREEUWENBERG, S. (2007b) Rule History and Versioning (Part 2). *Business Rules Journal*, Vol. 8, No. 12
- SPREEUWENBERG, S. (2008) Rule History and Versioning (Part 3). *Business Rules Journal*, Vol. 9, No. 1.
- TEDJASUKMANA, V. N. (2006) Translation of OCL Invariants into SQL:99 Integrity Constraints Hamburg, Germany
- VON HALLE, B. (2001) *Building a Business Rule System, Part 1. Data Management Review*,, Faulkner & Gray.
- VON HALLE, B. & RONALD, G. (2002) *Business rules applied: building better systems using the business rules approach*, New York, John Wiley & Sons.
- WEIDEN, M., HERMANS, L., SCHREIBER, G. & VAN DER ZEE, S. (2002) Classification and Representation of Business Rules. *European Business Rules Conference, June*.
- WEIDEN, M. R. (2000) A Critique of the Business-Rule Approach. *Department of Social Science Informatics*. Amsterdam, University of Amsterdam.
- ZACHMAN, J. A. (1987) A Framework for Information Systems Architecture. *IBM Systems Journal*, 26, 276-292.
- ZIMBRÃO, G., MIRANDA, R., SOUZA, J. M., ESTOLANO, M. H. & NETO, F. P. (2003) Enforcement of business rules in relational databases using constraints. *XVIII Simposio Brasileiro de Bancos de Dados/SBBD*

ZOET, M., VERSENDAAL, J., RAVESTEYN, P., WELKE, R., ZOET, M.,
VERSENDAAL, J., RAVESTEYN, P. & WELKE, R. (2011) ALIGNMENT OF
BUSINESS PROCESS
MANAGEMENT AND BUSINESS RULES. *European Conference on Information
Systems (ECIS) at AIS Electronic Library (AISeL).*
ECIS 2011

6. Anexos

Anexo 1

Anexo 1. Propuesta de un repositorio de reglas (Morgan, 2002e)

Tabla A1. Campos fundamentales del repositorio de reglas

	Nombre de Campo	Tipo de datos	Descripción
Atributo llave	ID	AutoNumérico	Identificador de la regla
	Versión	Texto	Número de la versión, en cualquier formato
	Último Cambio En	Fecha/Hora	Fecha de creación de la regla o del último cambio.
	Último Cambio Por	Texto	Nombre de la persona responsable
	Tipo	Número	Enlace al tipo de regla
	Identificador	Texto	Identificador para la regla independiente de la aplicación
	Nombre Descriptivo	Texto	Descripción textual de la regla
	Estado	Texto	Indicador para mostrar el estado actual de la regla.
	Notas	Memo	Campo para notas de format libre extensivo.
	Sentencia de Regla	Memo	Sentencia de reglas de negocio, en XML.
	Plantilla de Sentencia	Numérico	Enlace a la plantilla de regla.
	Sentencia Formal	Memo	Una representación formal de la regla.
	CategoríadeNegocio	Numérico	Enlace a la categoría de negocio.

Los datos que se guardan sobre el cambio de la versión de la regla pueden ser ampliados para brindar más detalles.

La plantilla de la regla o tipo de regla depende de las categorías de reglas adoptadas, para detallar información sobre las reglas se puede contar con otra tabla; para el

atributo estatus de la regla se proponen los valores revisados, aprobados y retirados, en dependencia de los estatus que se necesiten estos pueden ser modificados.

Para el atributo de autoría de la regla o responsable del último cambio se presenta un identificador para el autor de la regla, en caso que se desee pueden especificarse los identificadores de los diferentes tipos de autores y se definen en tablas adicionales los detalles sobre estos.

El diseño del repositorio puede tener en cuenta también las interrelaciones entre reglas, lo que es propuesto con las tablas relación regla-regla y la tabla auxiliar tipo de relación.

Tabla A2. Interrelación Regla-Regla

Nombre de Campo	Tipo de Datos	Description
IDRegla1	Numérico	Enlace a la regla 1
IDRegla2	Numérico	Enlace a la regla 2
RelationshipType	Numérico	Enlace al tipo de interrelación
FromDate	Fecha/Hora	Cuando la interrelación se convierte en efectiva
ToDate	Fecha/Hora	Cuando la interrelación expira.

Con la tabla tipo de relación explica el tipo de lectura para la interrelación entre las reglas, una estructura de esta tabla se propone como sigue.

Tabla A3. Lectura para interrelación entre reglas

Nombre de Campo	Tipo de datos	Descripción
ID	AutoNumérico	Identificador único para los tipos de interrelación.
Nombre de la Interrelación	Texto	Nombre del tipo de interrelación
Lectura desde Regla1	Texto	La forma de ver la interrelación desde la regla 1
Lectura desde Regla2	Texto	La forma de ver la interrelación desde la regla 2

Lectura desde Regla1, describe cómo se lee la relación de la regla 1 con respecto a la regla 2.

Lectura desde Regla 2, describe como se lee la relación desde la regla 2 con respecto a regla 1.

Si existen múltiples interrelaciones esta estructura puede extenderse.

En la propuesta de Morgan se incluye una manera flexible de representar los contenedores de las reglas, que pueden ser módulos de software, componentes de una máquina de regla u otro, cuyas formas identifican por tiposde Realizaciones. Cada realización es entendida dentro de un contenedor notificable, que se puede representar como en la figura siguiente:

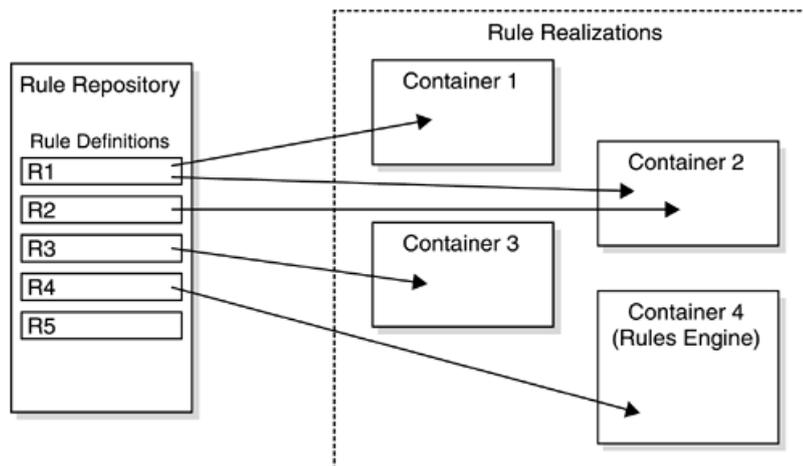


Figura A1. Enlace entre definiciones de las reglas y sus realizaciones.