

TRADUCTOR DE REGLAS DE NEGOCIO DE LENGUAJE TÉCNICO LPT A LENGUAJE FORMAL SQL

Martha Beatriz Boggiano Castillo*, Ariel Calderón Valdés*, Alain Pérez Alonso, Ramiro Pèrez Vazquez.

mbeatriz@uclv.edu.cu, acaleron@uclv.edu.cu, apa@uclv.edu.cu, rperez@uclv.edu.cu

Universidad Marta Abreu de Las Villas.

RESUMEN

El desarrollo de sistemas de información siguiendo el *enfoque de reglas de negocio* es una tendencia actual en las ciencias de la Computación. Las reglas o políticas del negocio son capturadas por el especialista del negocio y finalmente deben ser generadas automáticamente de alguna manera que el sistema de información pueda utilizarlas. Con este trabajo se propone una arquitectura solución al problema de implementar automáticamente las reglas de negocio en forma de recursos de bases de datos. Se refina y extiende el lenguaje LPT (lenguaje de patrones técnicos) para expresar reglas de negocio, se diseña e implementa un traductor en software libre que aceptando como entradas un repositorio de reglas en LPT y el catálogo de la base de datos, ambos en formato XML, obtiene la implementación de las reglas en forma de disparadores (triggers) y vistas, para ser usados de dos formas diferentes por los sistemas de información, de manera inmediata o mediata, cuando las operaciones sobre la base de datos provoquen que se infrinjan las reglas del negocio registradas.

Palabras clave: reglas de negocio, reglas en lenguaje técnico, reglas en SQL,

1. INTRODUCCION

Existen múltiples tipos de reglas de negocio. Debido a su diversidad y complejidad los autores tienden a agruparlas y clasificarlas siguiendo diferentes puntos de vista, pero siempre con el objetivo común de recoger todas las reglas del análisis de requisitos del negocio. Fueron analizadas las clasificaciones según Morgan [11], Weiden [15], Ashwell [3] y Solivares [5].

De acuerdo con Morgan [11] las reglas de negocio pueden expresarse de diversas formas, principalmente, según su especificación en el desarrollo de un sistema de información o incluso de acuerdo a la manera en que se introduzcan a este. Fundamentalmente, se definen tres niveles de expresión de las reglas de negocio:

- ❖ Informal: este nivel proporciona una sentencia en lenguaje natural, sin un rango limitado de parámetro, tal y como el cliente del negocio desee.

- ❖ Técnico: este nivel combina referencias a datos estructurados, operadores y restricciones con el lenguaje natural, nivel intermedio entre la entrada de la regla y su implementación.
- ❖ Formal: este nivel proporciona sentencias conforme a una sintaxis definida y proporciona la funcionalidad automática de la regla.

Algunos autores como Morgan [11], Weiden [15] presentan los patrones de reglas para reflejar las formas o tipos de problemas que un sistema automatizado pretende negociar. Morgan relaciona varios tipos de reglas como las de Restricción, Clasificación, Cálculo y Enumeración. De estos son de interés en este trabajo los que pertenecen al tipo de restricción.

Para este trabajo se planifica

El patrón para reglas de negocio tipo Restricción es el común entre los patrones de reglas de negocio y establece una restricción sobre el sujeto de una regla. El patrón propuesto por Morgan es el siguiente:

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

Elemento	Significado
<determinante>	Es el determinante para cada sujeto, por ejemplo: Una, Uno, El, La, Cada, Todos. Según el mejor sentido en la redacción.
<sujeto>	Es un elemento de la Base de Datos del negocio, tal como entidades y objetos. La entidad puede ser cualificada por otros atributos descriptores, tales como la existencia en un estado particular o relacionada con una aplicación específica de la regla.
<características>	Describe las características del sujeto en el negocio, tanto internas como relacionadas con otras entidades.
<hechos>	Hechos relativos al estado o comportamiento de la Base de Datos del negocio, incluyendo o no al sujeto.

Tabla 1. Elementos del Patrón de Restricción de Morgan.

En [1, 2] este patrón fue modificado con el fin de mejorarlo y ampliarlo , obteniéndose como resultado el siguiente patrón:

<determinante> <sujeto> (no puede tener <características>) | (puede tener <características> solo si <hechos>).

El significado de los elementos del patrón no sufrió grandes cambios, como se muestra en la Tabla 2.

Elemento	Significado
<determinante>	Es el determinante para cada sujeto, por ejemplo: Una, Uno, El, La, Cada, Todos. Según el mejor sentido en la redacción.
<sujeto>	Es una tabla en la Base de Datos del negocio o una clasificación de la misma.
<hecho>	Hechos relativos al estado o comportamiento de la Base de Datos del negocio, incluyendo o no al sujeto.
<característica>	Describe las características del sujeto en el negocio, tanto internas como relacionadas con otras entidades. Pueden incluir hechos con el fin de caracterizar al sujeto.

Tabla 2. Elementos del Patrón de Restricción modificado.

Para la representación formal de las reglas de negocio se usan los diferentes lenguajes de programación, los scripts, en la Web, también el SQL. Morgan[11].

El lenguaje SQL estándar en sus múltiples revisiones y particularmente la implementación para SQL Server 2005 [12] ofrece varios recursos de manejo de datos ideales para la implementación de reglas de negocio. En particular para implementar reglas de negocio tipo Restricción son de gran utilidad los triggers, que son asociados a las tablas en las que puede violarse la regla, y las vistas, que almacenan la información de los sujetos que violan una determinada regla.

Restricciones (Constraints) CHECK:

Las restricciones CHECK exigen la integridad del dominio mediante la limitación de los valores que puede aceptar una columna. Se puede crear una restricción CHECK con cualquier expresión lógica (booleana) que devuelva TRUE (verdadero) o FALSE (falso) basándose en operadores lógicos [13]. 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 [12].

Desencadenadores (Triggers):

Un trigger es una pieza de código ejecutable, que consiste de instrucciones declarativas y procedurales y que se almacenan en el catálogo del SGBD[9]. Son una clase especial definida para la ejecución automática al emitirse una instrucción UPDATE, INSERT o DELETE en una tabla o una vista. Son una herramienta eficaz que pueden utilizar los sitios para exigir automáticamente las reglas comerciales cuando se modifican los datos. Amplían la lógica de comprobación de integridad, valores predeterminados y reglas del estándar SQL, aunque se deben utilizar las restricciones y los valores predeterminados siempre que estos aporten toda la funcionalidad necesaria [10].

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 [10]. 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, dado que los usuarios solo pueden ver los definidos en la vista y no los que hay en la tabla subyacente [12].

Transacciones:

Una transacción es una unidad única de trabajo. Si una transacción tiene éxito, todas las modificaciones de los datos realizadas durante la transacción se confirman y se convierten en una parte permanente de la base de datos. Si una transacción encuentra errores y debe cancelarse o revertirse, se borran todas las modificaciones de los datos [8]. Una transacción es una secuencia de operaciones realizadas como una sola unidad lógica de trabajo. Una unidad lógica de trabajo debe exhibir cuatro propiedades, conocidas como propiedades de atomicidad, coherencia, aislamiento y DURABILIDAD (ACID), PARA SER CALIFICADA COMO TRANSACCIÓN.

2. DESARROLLO

En este trabajo se adoptó la clasificación ofrecida por Morgan debido a que define patrones de reglas más elaborados y bien estructurados.

De la semántica de los elementos del patrón es necesario destacar la ampliación realizada a las <características> con el objetivo de representar una mayor cantidad de sentencias

2.1. Lenguaje de Patrones Técnicos

El Lenguaje de Patrones Técnicos (LPT) no es más que una expresión matemática de los elementos que conforman los patrones de regla. Para este fin el lenguaje posee una notación específica, operadores lógicos, operadores aritméticos y funciones, que juntos conforman una potente herramienta para aceptar múltiples reglas expresadas en un lenguaje cercano al natural. Este lenguaje fue creado en [1] y extendido en [2] y en este trabajo. Su creación como un lenguaje técnico y evolución posterior ha sido guiada por la necesidad de aceptar variantes cada vez más complejas, preservando siempre una estructura sencilla. Esta estructura le brinda dos ventajas fundamentales:

- ❖ La simple transformación desde un lenguaje seminatural.
- ❖ Creación relativamente fácil por un especialista técnico del negocio.

2.2. La Notación Punto.

Esta característica del LPT más que una notación es un estilo. Su existencia es imprescindible, pues establece el medio de acceso a los atributos de tablas y posibilita la navegación, lo cual refiere Zimbrão [16]. Esta notación le brinda consistencia al lenguaje.

Acceso simple a un atributo:

Tabla 1 [. Atributo]

Camino de navegación entre entidades:

Tabla 1. Tabla 2. (...) . Tabla N [. Atributo]

En ambos casos es necesario destacar la opción de terminar o no con un atributo en la última tabla. Si concluyera solamente con el nombre de una tabla entonces se está haciendo referencia las instancias de la tabla como tal, específicamente a su(s) atributo(s) identificador(es).

Al utilizar la notación punto se tiene como resultado elementos individuales o elementos múltiples. Los primeros son útiles para el acceso a atributos específicos de una tabla.

Al expresar en lenguaje natural: “El paciente de la evolución...” (Evolución. Paciente) se está refiriendo a un único paciente, pues al ser la relación uno a muchos, se tiene que una evolución solo posee un paciente.

La notación punto no es el único modo de obtener elementos individuales ya que estos pueden ser el resultado de operadores de conjuntos, los cuales se verán más adelante, o simplemente constantes.

Los elementos múltiples surgen cuando a partir de la notación punto se obtienen como resultado varios elementos, por ejemplo, al indicar “Las Evoluciones del Paciente...” (Paciente. Evolución) se conciben varias evoluciones relacionadas con un único paciente, como muestra la cardinalidad del diagrama. Nótese que el orden de la notación es muy importante.

En estudio previo [14] se extiende la posibilidad de navegar por relaciones entre entidades de cualquier cardinalidad. Por tanto un elemento múltiple es únicamente resultado de una navegación en la cual al menos se tenga una interrelación (M:M) o (1:M). Igualmente se distingue la obtención elementos simples, aunque un elemento es simple cuando no es múltiple.

2.3. Operadores lógicos, aritméticos y de comparación del LPT.

Los operadores utilizados por el LPT son los básicos para construir sentencias con un alto grado de complejidad. Algunos rechazan el uso de estos por considerar que las reglas deben ser siempre extremadamente sencillas [11], pero la realidad demanda en no pocos casos reglas con una lógica compleja. A continuación se muestran los mismos.

Tipo	Operadores
Lógicos	OR,AND,NOT,XOR
Aritméticos	+, -, *, /
Comparación	<, >, <=, >=, <>, =

Conjunto EXIST,EMPTY,SIZEOF,MIN,MAX,AVG,SUM,SIZEOFDIF,AVGDIF,SUMDIF

Tabla 3 Operadores del LPT.

2.4. Enfoques para la implementación de reglas de negocio tipo restricción

Luego de un cambio de estado de la base de datos del negocio (provocado por una operación de inserción, modificación o actualización) se inicia una revisión de todas las reglas tipo restricción asociadas a este cambio.

El método de chequeo *inmediato* consiste en que inmediatamente de finalizada la operación, si se viola alguna regla, se lanza un mensaje de error y se deshace la operación. Este método de chequeo inmediato es brevemente abordado por Date [7] al definir su Regla de Oro y en general pertenece a vías tradicionales de implementación de reglas de negocio [4]. Para implementar este proceso basta con utilizar como recursos de bases de datos *triggers* y vistas para el caso de reglas complejas y *constraints* para el caso de reglas simples.

En ocasiones cuando se buscan violaciones de reglas de negocio se necesita analizar más allá de una operación en particular y verlas como ilegalidades de un conjunto de operaciones [6, 7]. El método de chequeo *mediato* ocurre cuando no se toman acciones inmediatamente después de ser violada la regla, sino que estas reglas son consideradas como “posibles reglas infringidas”. Luego de concluido el conjunto de operaciones se extraen las reglas que han realmente han sido violadas de entre las “posibles reglas infringidas” y con esta lista el sistema del negocio puede decidir qué hacer, si deshacer todas las operaciones, analizar las reglas que han sido violadas y de cada una cuales son los objetos causantes de la infracción. Con esta información se puede hacer un análisis profundo antes de tomar una medida, e incluso en casos extremos buscar más detalles. La decisión a tomar puede ser en caliente preguntándole directamente al cliente ¿qué desea hacer? o realizar acciones automáticas predefinidas.

Bajo este enfoque las políticas prohibitorias son asociadas a un conjunto de operaciones y no a una única operación como tradicionalmente se llevaba a cabo [6]. La base esencial para

implementar este proceso ha de ser otro de los recursos que brindan la mayoría de los gestores, definido en el estándar SQL: las transacciones [7].

Es necesario destacar que el Sistema de Información es el responsable de tomar la decisión de cuál enfoque utilizar.

2.5. Proceso de traducción de las reglas

En la Figura 1 se muestra el esquema general del proceso de generación de las reglas. Se parte del *Repositorio de Reglas* donde está almacenada la regla en lenguaje natural y técnico (en LPT). La entrada del traductor es la regla en LPT que se extrae del *Repositorio de Reglas*. Durante el proceso de traducción se consulta la información del catalogo que está almacenada en el *Repositorio de Información del Catálogo*. Este repositorio contiene la información de las tablas, atributos, triggers, vistas y funciones que están implementadas en la base de datos física. Posteriormente se genera la información necesaria para generar la regla que es almacenada en el *Repositorio de Generación*. Esta información consiste en la consulta SQL base y la lista de eventos, esto es los eventos que posibilitan un cambio de estado relacionado con cierta regla. Vale destacar que dicha información es suficiente para generar la regla tipo restricción desde cualquiera de los dos enfoques analizados: mediato o inmediato. Finalmente se genera la regla a partir de la información extraída del *Repositorio de Generación* de acuerdo al enfoque que se maneje, y se actualiza el *Repositorio de Reglas* obteniéndose finalmente la representación de la regla en lenguaje natural, técnico y formal.

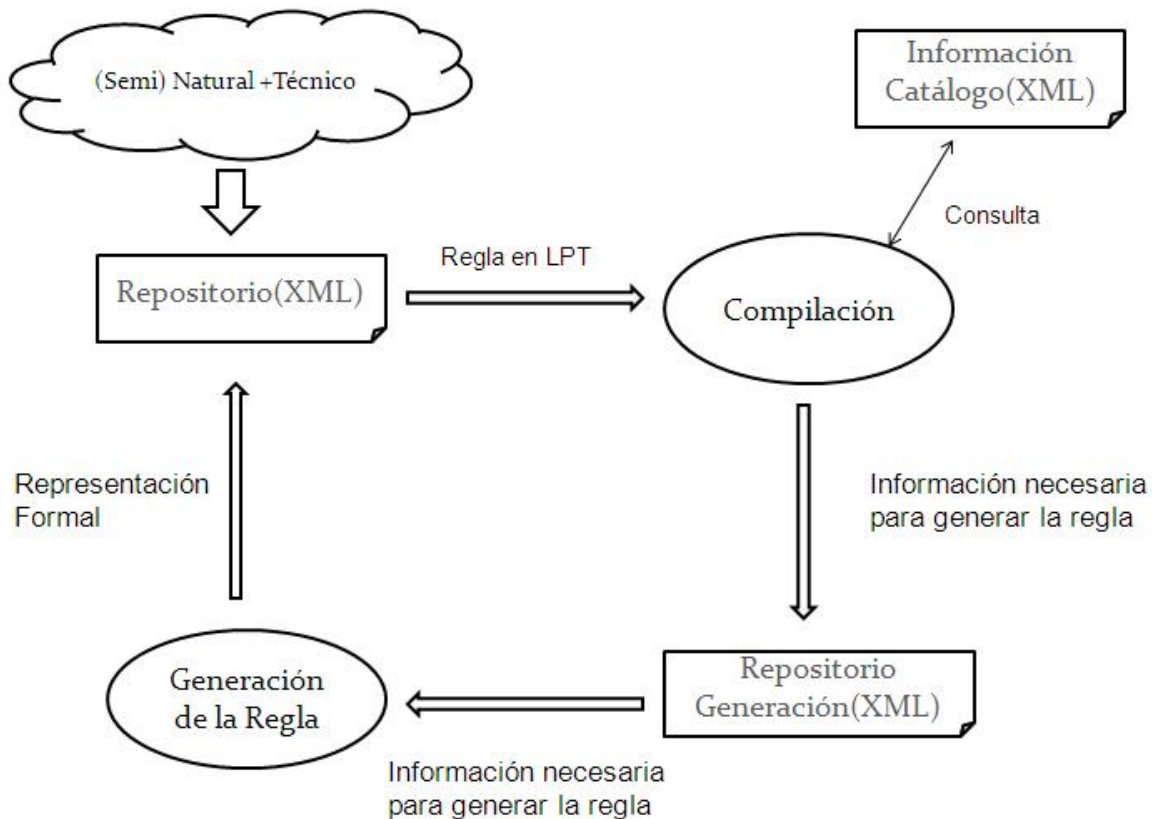


Figura 1 Esquema general del proceso de traducción.

2.6. Generación de una regla de negocio tipo restricción.

A continuación se muestra un ejemplo de generación de una regla de negocio tipo restricción. Partimos de la siguiente regla expresada en LPT:

Un paciente no puede tener sizeof (sujeto.ExamenFisico.idexamenfisico)>8

Luego del proceso de compilación se obtiene la información necesaria para implementar la regla tal y como se muestra a continuación:

Consulta SQL base:

```
(SELECT COUNT ( b.idExamenFisico ) FROM Paciente a , ExamenFisico b WHERE
(sujeto.idPaciente=a.idPaciente) AND (a.idPaciente=b.idPaciente)) > 8
```

Lista de Eventos:

- *INSERT ON ExamenFisico*

Finalmente se genera la regla, en este caso tomando en cuenta el enfoque mediano y en MS SQL Server. El código generado se muestra a continuación:

```
CREATE TRIGGER TIRN#6 ON ExamenFisico FOR INSERT AS if (EXISTS(SELECT  
* FROM VRN#6 ) ) BEGIN raiserror(RN#6,16,1); rollback transaction; END
```

```
CREATE VIEW VRN#6 AS (SELECT * FROM Paciente sujeto WHERE ((SELECT  
COUNT( b.idExamenFisico ) FROM Paciente a , ExamenFisico b WHERE  
(sujeto.idPaciente=a.idPaciente) AND (a.idPaciente=b.idPaciente)) > 8))
```

Nótese que la consulta base constituye el cuerpo de la vista, mientras que la cantidad de elementos en la lista de eventos va a determinar el número de triggers que deben ser creados.

3. CONCLUSIONES

Con el desarrollo de este trabajo obtienen extensiones al lenguaje LPT, con el correspondiente soporte informático capaz de convertir reglas de negocio expresadas en LPT a reglas expresadas en SQL, usando triggers y vistas, de manera que un sistema de información pueda procesar las reglas con un enfoque inmediato o un enfoque mediano.

Usando dos repositorios físicos de las reglas con formato XML, y el catálogo de la base de datos también extraído a un fichero XML, la herramienta es capaz de generar automáticamente la implementación de las reglas. Las reglas tipo restricción, involucradas en cualquier evento sobre la base de datos son generadas aquí para el gestor MS SQL Server; el módulo está programado sobre plataforma libre, en Java.

REFERENCIAS

- [1]. Alonso, A.P., *Aplicación para reglas de restricción en negocios*, in *Departamento de Bases de Datos*. 2008, Universidad Central de Las Villas: Santa Clara.
- [2]. Alonso, A.P., *Reglas de Negocio en Bases de Datos Relacionales*, in *Departamento de Bases de Datos*. 2010, Universidad Central de Las Villas: Santa Clara.
- [3]. Ashwell, R., *Define Business Rules*. 2006, CRaG Systems. p. 7.
- [4]. Bajec, M., R. Rupnik, and M. Krisper (2000) *USING BUSINESS RULES TECHNOLOGIES TO BRIDGE THE GAP BETWEEN BUSINESS AND BUSINESS APPLICATIONS*. 8.
- [5]. Besembel, I.M. and E. Chacón *Objetos y reglas de negocios en la integración y automatización de procesos de producción continua*. 12.
- [6]. Choi, E.-H., T. Tsuchiya, and T. Kikonu, *Model Checking Active Database Rules* 2006 Research Center for Verification and Semantics (CVS) National Institute of Advanced Industrial Science and Technology (AIST) Nakouji, Amagasaki, Hyogo. p. 20.
- [7]. Date, C.J., *Constraints & Predicates: A Brief Tutorial (Part 2)*. Business Rules Journal, 2001. V2.
- [8]. Gennick, J., *SQL pocket guide*. 2 ed. 2006: O'Reilly Media. 184.
- [9]. González, L., *Sistemas de Bases de Datos postrelacionales.Triggers* 2010.
- [10]. Melton, J. and A.R. Simon, *SQL1999: understanding relational language components*. 2002: Morgab Kaufmann.
- [11]. Morgan, T., *Business Rules and Information Systems: Aligning IT with Business Goals*. 2002, Addison Wesley. p. 384.
- [12]. MSDN, *MSDN LIBRARY VisualStudio 2008*. 2008, Microsoft Corporation.
- [13]. Oppel, A. and R. Sheldon, *SQL: a beginner's guide*. Vol. 3. 2008: McGraw-Hill Profesional.
- [14]. Toledo, A.P., *Solución al problema de la cardinalidad en la generación automática de reglas de negocio en bases de datos relacionales.*, in *Departamento de Bases de Datos*. 2009, Universidad Central "Marta Abreu" de Las Villas: Santa Clara. p. 92.
- [15]. Weiden, M., et al. (2002) *Classification and Representation of Business Rules*.
- [16]. Zimbrão, G., et al. (2002) *Enforcement of Business Rules in Relational Databases Using Constraints*. 13.