Seventh LACCEI Latin American and Caribbean Conference for Engineering and Technology (LACCEI'2009)
"Energy and Technology for the Americas: Education, Innovation, Technology and Practice"
June 2-5, 2009, San Cristóbal, Venezuela.

Traducción De Un Patrón De Reglas De Negocio En Bases De Datos Relacionales

Msc. Martha Beatriz Boggiano Castillo,
Universidad Central de las Villas, Santa Clara, Cuba, **mbeatriz@uclv.edu.cu**

Lic Alaín Pérez Alonso.
Universidad Central de las Villas, Santa Clara, Cuba, **apa@uclv.edu.cu**

Dr. Ramiro Pérez Vázquez
Universidad Central de las Villas, Santa Clara, Cuba, **rperez@uclv.edu.cu**

Msc. María Elena Martínez Del Busto
Universidad Central de las Villas, Santa Clara, Cuba. **mmbusto@uclv.edu.cu**

Dra. Luisa González González
Universidad Central de las Villas, Santa Clara, Cuba. **luisagon@uclv.edu.cu**

RESUMEN

Los problemas en la concepción de Sistemas de Información, debido al tratamiento informal de los requerimientos, han permitido considerar como factor de esencial importancia la forma en que se manejan los requisitos, enfocados actualmente como reglas de negocio.

El propósito de este trabajo es generar automáticamente las reglas de restricción en un sistema de información, específicamente aquellas que puedan residir en la base de datos ganando con esto agilidad, transparencia y reducción del costo en un negocio. Se usa un patrón para estas reglas, analizando los recursos que posibiliten su implementación en una base de datos, a un nivel técnico Se modifica un patrón de reglas adaptándolo a las nuevas exigencias de las reglas, se eligen los recursos de bases de datos para implementarlas, se construye un compilador que genere el código de las partes del patrón adicionando un generador de los mecanismos de base de datos correspondientes.

Palabras claves: Reglas de Negocio, generar código automáticamente, reglas de negocio de restricción

ABSTRACT

The problems in the design of information systems, due to the informal treatment of the requirements, have permitted to consider it as an essential factor the way that requirement are handled, currently focused business rules.

The purpose of this work is to generate restricted rules automatically to a system of information, specifically those that may reside in the database to gain from this flexibility, transparency and cost reduction in a business. It describes some benefits of using business rules. It used a pattern to these rules, analyzing the resources that will enable it implementation in a database and determine it form of storage. It is modified a pattern of rules adapted to the new requirements of the rules, It is chosen the resources database to implement them, and it is built a

compiler that generates this code of pattern parts, it adding a generator of database mechanism that correspond with this pattern code.

Keywords: Business rules, generates code automatically, restricted business rules

1. INTRODUCTION

Actualmente muchos especialistas en Sistemas de Información consideran que el inicio de la identificación de los requisitos son las reglas de negocio. Estas reglas pueden ser consideradas como sentencias que permiten a los usuarios expertos definir políticas, condiciones y conocimientos del negocio en unidades pequeñas y aisladas. Parte de los requisitos, enfocados hoy como reglas de negocio, tradicionalmente han sido realizados "a mano" por el programador en el código fuente del programa, en los objetos de bases de datos de la aplicación, o en ambos.

Adicionando la automatización a este proceso se evitan fallas y, de este modo, se logra importantes reducciones de tiempo y costo. Para automatizar estos "requisitos" es necesario un sistema encargado de darles funcionalidad, logrando implementarlas en un lenguaje y mantenerlas almacenadas para su manipulación.

Estudios actuales tienden a almacenar las reglas en un repositorio, que son manejadas por un motor de reglas independiente al Sistema de Información del negocio. Por tanto las restricciones no se codifican en el sistema y este sólo puede realizar accesos necesarios al repositorio.

La idea de las reglas de negocio, como definición de la política deseada, podría proporcionar una solución por su posible tratamiento en bases de datos y los beneficios que proporcionan.

Varios son los beneficios que pueden derivarse del uso de Reglas de Negocio. De todos, según Lowenthal (Lowenthal, 2005) los tres más importantes son:

Agilidad: respuesta simple y rápida a los requisitos dinámicos.

Reducción del Costo: bajo costo para crear o actualizar las partes de aplicaciones que implementan las políticas del negocio. Transparencia: las reglas permiten fácilmente la auditoría que los servicios de software llevan a cabo en sus políticas de negocios correspondientes.

2. IMPLEMENTACIÓN DE REGLAS DE NEGOCIO

Existen muchos tipos de reglas de negocio. Debido a su diversidad y complejidad los autores tienden a agruparlas y clasificarlas siguiendo diferentes puntos de vista, entre estas tenemos la clasificación según Solivares, Lowenthal, Morgan (Pérez, Boggiano 2007).

Se realizó un análisis de las clasificaciones de estos autores y se adoptó la clasificación de Morgan por considerarl la más apropiada por tener en cuena un patrón de regla de clasificación más elaborado.

Según Tony Morgan (Morgan, 2002), la forma más conveniente de crear sentencias de regla es seleccionar patrones adecuados desde una pequeña lista disponible. Por ejemplo: <sentencia>:= <sujeto> debe <restricción>

Morgan sostiene que no existe un estándar de cómo hacer reglas de negocio pero sí es posible hacer algunas recomendaciones. Los patrones pueden 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 a los tipos de restricciones .

Las reglas de restricción son aquellas restringen los datos contenidos en el sistema, estas . pueden solaparse en cierto modo con las reglas del modelo de datos, pues aquellas también impiden la introducción de datos erróneos. La diferencia estriba en que las reglas de restricción condicionan el valor de los atributos o propiedades de una entidad más allá de las restricciones básicas sobre las mismas existentes.

2.1 ADOPTANDO UN PATRÓN PARA LAS REGLAS DE NEGOCIO.

Las reglas se pueden implementar de formas disímiles e incluso pueden existir diferentes técnicas para una misma regla. Morgan (Morgan, 2002) muestra algunas de las posibles a utilizar: Lenguajes de Programación, Scripts, Bases de Datos.

Diferentes reglas pueden ser aplicadas a diferentes capas de la arquitectura. Las reglas pueden ser aplicadas a: las GUI, WebServer, Aplicaciones, una capa intermedia y a las bases de Datos. Las reglas pueden ser representas por diferentes mecanismos de implementación, simples validaciones en las GUI, complejos chequeos en la capra de aplicación, procedimientos almacenados, y trigger en las bases de datos. Arsanjani (Arsanjani, 2001) propone patrones para objetos, de regla para construir reglaspara lenguajes de programación, usa usa patrones de diseño como "composite", "cluster". Este no es de interés en este caso.

Hemos constatado que muchas de las reglas de negocio encajan más naturalmente dentro de la base de datos, donde pueden tener un contacto más directo con los datos (Mota, 2005). Los tipos de tecnología más usados son Oracle, SQL Server y Postgres. Otros productos de base de datos correlativos tienen ampliamente capacidades similares, por lo que no es nada complejo extender esta investigación a otros lenguajes. En el caso del Servidor SQL, puede programarse en un dialecto llamado Transact-SQL.

Lo más habitual es usar las reglas de negocio respecto a palabras funcionales que estén alrededor de lo básico (CREATE, READ, UPDATE y DELETE). Estos mecanismos son comunes a todos los servicios de datos:

Restricciones (Constraints) CHECK, Desencadenadores (Triggers), Vistas, Funciones definidas por el usuario y las Assertions:

Aquí se trabaja con los desencadenadores y las funciones para implementar las reglas de restricciones en la base de datos del negocio.

Entre los patrones estudiados para representar reglas de negocio, fue de interés el propuesto por Morgan (Morgan, 2002). (Arsanjani, 2001)

como El patrón propuesto por Morgan para regla de restricción básica se tiene:

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

Este es el más común entre los patrones para reglas de negocio, establece una restricción sobre el sujeto de una

Regla

Tabla 1 Elementos del Patrón.

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.

Se decidió realizar modificaciones en función de mejorarlo y adaptarlo a la necesidad de obtener efectos palpables. Resultando el patrón de restricción que sigue (manteniendo sus convenios):

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

Los convenios utilizados para los patrones en este trabajo son los siguientes:

- Paréntesis ( ) Encierran un grupo de ítems.
- Corchetes [ ] Encierran elementos opcionales.
- Barra Vertical | Separa términos alternativos.
- Corchetes Angulares <> encierran términos especiales como los mostrados en la siguiente tabla.

2.2 UN CASO DE RESTRICCIÓN BÁSICA Y LISTA DE RESTRICCIONES.

Se presenta una regla de negocio para un problema de transplante renal.

Lenguaje Natural:

Un Paciente no puede tener Donantes Potenciales con más de 4 Exámenes Físicos o tener en su Evolución una temperatura máxima mayor de 42 grados.

Para lograr reglas sencillas y fáciles de mantener, tratando de separarlas en reglas más pequeñas, según sea posible, se tiene

Lenguaje Natural:

Un Paciente no puede tener Donantes Potenciales con más de 4 Exámenes Físicos.

Un Paciente no puede tener una temperatura máxima de 42 grados.

Esta manera de expresar esta regla muy cercana al cliente se le denomina en Lenguaje Natural, teniendo en cuenta el patrón adoptado. Con el objetivo de lograr una implementación SQL, se considera una representación intermedia, denominado lenguaje técnico, para representar la regla en lenguaje técnico se usa la representación de la regla usando navegación punto (Morgan, 2002)

Este estilo es utilizado en el lenguaje técnico para acceder a los atributos de las entidades y navegar en las relaciones entre estas.

Acceso simple a un atributo:

Entidad_1.Atributo

Navegar entre relaciones de entidades:

Entidad_1.Entidad_2. (…) .Entidad_n.Atributo

Por ejemplo, si se desea acceder al nombre del paciente se puede alcanzar anotando Paciente.Nombre, pero también es posible navegar en las relaciones de la Figura 1.

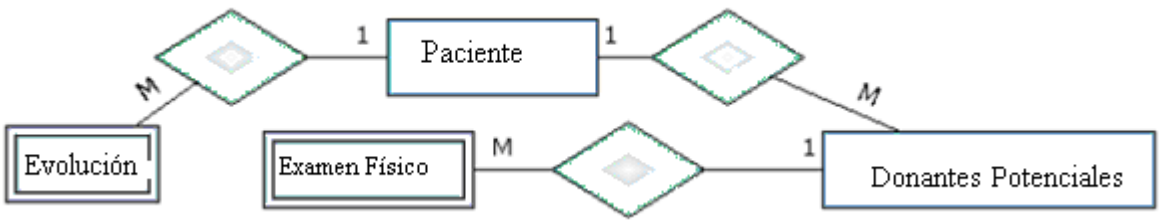

Figura1. Porción del Esquema E/R para Transplante Renal.

Al utilizar la navegación de la notación punto para establecer relaciones entre entidades y acceder a los atributos de estas, se obtendrán dos tipos de resultados principales debido a la cardinalidad entre los objetos del negocio; pues como mismo se puede especificar un solo miembro del negocio, también es viable hacer referencia a un conjunto de estos y resulta necesario discernir ambos casos.

Múltiples elementos: Al indicar "Las Evoluciones del Paciente…" (Paciente.Evolucion) se conciben varias evoluciones relacionadas con un único paciente como muestra la cardinalidad del diagrama. Nótese nuevamente que el orden de la notación es muy importante.

Una expresión de regla en lenguaje técnico sería:

Un Paciente no puede tener sizeof(Evolucion.idEvolucion) > 30

<Sujeto>: Paciente

<Características>: sizeof(Evolucion.idEvolucion) > 30

A esta regla le corresponde en lenguaje natural la siguiente:

Un paciente no puede tener más de 30 evoluciones.

2.3 EL COMPILADOR O INTÉRPRETE DE REGLAS.

Las reglas en lenguaje técnico deben ser automatizadas, para convertir estas reglas en triggers y función, expuestos en el Capítulo II, se crea un compilador, el que va a ser utilizado por una aplicación que ha de permitir editar la regla por un analista. En la figura siguiente se muestra el caso de uso **Compilar Regla**. En la cual como se puede apreciar es el editor de reglas el que emplea al compilador.

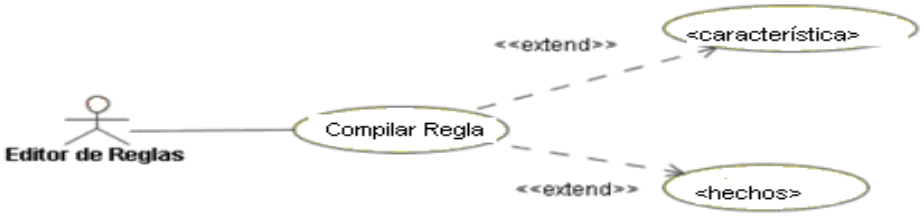

Figura 2. Caso de uso del compilador

2.4 EL PROCESO DE INTERPRETACIÓN DE OPERADORES Y FUNCIONES.

El proceso de compilación, que realmente es un traductor de las Reglas de Negocios, tipo restricción, utilizado en la versión actual de un editor de reglas en lenguaje técnico se hace en una sola pasada. Durante este proceso se va generando directamente parte del código resultante, el resto de este se formaliza en la aplicación.
Este compilador está compuesto por otros dos compiladores, uno se encarga de compilar las <características> y otro los <hechos>. El proceso de compilación se muestra mediante el diagrama de actividad (Figura 3):

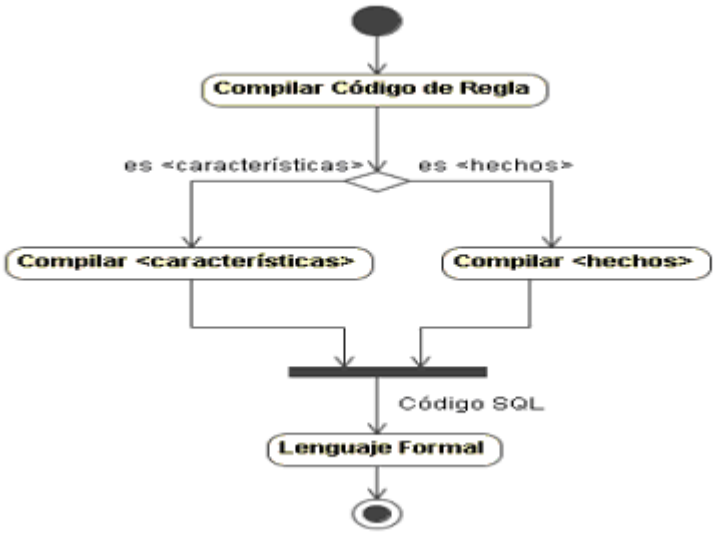

Figura 3. Diagrama de Actividad en la Compilación o Traducción

El compilador fue realizado en el software Parser Generator v1.12 con Lex y YACC para Delphi. Con Lex es se crean patrones de coincidencia en códigos. Los patrones son especificados usando expresiones regulares, cada una de estas tiene una acción correspondiente asociada a ellos. Cuando se encuentra una cadena que coincide con la expresión regular se realiza la acción respectiva.

Con YACC ,dada la especificación BNF (Forma Normal de Backus) de la gramática, puede generar el parser correspondiente. YACC no acepta cada gramática presentada a él; sin embargo, la clase de gramática que acepta es generalmente poderosa, más que suficiente para las necesidades de la programación (Bumble-Bee, 2000).

En el Lex se definen principalmente los tipos de datos, los operadores y las funciones que se aceptan del lenguaje técnico, estos son útiles para conformar la gramática de la regla, tipo restricción.

Solamente se generan triggers y funciones. Se trabaja en mejorar este compilador representar restricciones CHECK, realizar un chequeo riguroso de tipo y hacer consideraciones sobre la repercusión de la transformación de las reglas de negocio en la base de datos del negocio.

Actualmente este software se aplica para un conjunto de reglas de estricción para el problema de transplante renal en una base de datos SQL Server.

2.5 RESULTADOS DEL PROCESO DE COMPILACIÓN O TRADUCCIÓN DE LOS PATRONES DE REGLAS.

Aplicando patrones de reglas de restricción en una base de datos para trasplante renal

Para el Ejemplo :

Un nefrólogo puede atender más de 12 pacientes solo si la cantidad de Pacientes es mayor que 50.

Se utiliza una aplicación que al usar el traductor diseñado y programado, genera en una base de datos SQL Server , del negocio de transplante renal la función y el trigger siguiente:

La función:

```
CREATE FUNCTION FRN#0(@idSujeto Integer)
Returns BIT AS
begin
declare @result BIT
set @result = 0
if (
```

```
(SELECT COUNT(b.idpaciente)
FROM Nefrologo a, Paciente b
Where (@idSujeto=a.idNefrologo) and (a.idNefrologo=b.idNefrologo)) > 12)
if not (
(SELECT COUNT(a.idpaciente)
FROM Paciente a
Where ) > 50)
set @result = 1
return @result
end

CREATE Trigger TRN#00 ON Paciente
AFTER INSERT,UPDATE
AS
declare @Subject int
set @subject =
(SELECT a.idNefrologo
FROM Nefrologo a, Inserted b
Where (a.idNefrologo=b.idNefrologo))
if dbo.FRN#0(@Subject) = 1
begin
 raiserror('RN#0',16,1);
 rollback transaction;
end
```

Estos triggers, como se ha podido observar, declaran un @subject que es el <sujeto> condicionado de la regla, el que se extrae a través de una consulta definida por la notación punto. Después en una condicional se cuestiona si la regla es quebrantada o no para el sujeto específico al cual se le realizó una modificación relacional, en caso afirmativo se toman las medidas correspondientes.

La función es el cuerpo de la regla. Los trigger harán referencia a una función. Esta es la encargada de realizar el cuerpo de la regal, en esta se verifica si el sujeto viola las condiciones impuestas por la restricción. Su objetivo es responder si se infringe alguna de las prohibiciones mediante un valor booleano.La función FRN#0 es llamada desde el disparador o trigger

3. CONCLUSION

En este artículo se ha abordado la necesidad de tratar los requisitos del sistema usando el enfoque de reglas de negocio, mostrando su preponderancia en el proceso de manejar la organización, de modo que se pueda contar con un potencial para una reducción drástica en el número de errores.

Se consideran las reglas de negocio clasificadas de tipo restricción, formalizada en un patrón de regla de negocio, a partir del propuesto por Morgan . Se construye y usa un programa que compila o traduce estas regla a partir de una entrada de la regla en lenguaje técnico y usando la navegación punto que tiene como entrada la regla en el patrón adoptado y haciendo consideraciones para cada parte de este patrón genera automáticamente un trigger de inserción y actualización que implementa una regla de restricción para una Base de Datos SQL Server 2000 de un sistema de información, se presentan ejemplos para una base de datos de transplante renal. Siendo válido este compilador para cualquier tipo de negocio, que use una base de datos del negocio en SQL Server 2000.

La automatización para generar reglas de negocio permite evitar fallas, de modo que hay importantes reducciones de tiempo y costo.

REFERENCIAS

Ali Arsanjani (2001). Rule Object 2001: A pattern Language for Adpative and Scalable Business Rule Contruction.

Bajec, M.K., Marjan *Managing business rules in enterprises* 2006, Faculty of Computer and Information Science, University of Ljubljana: Ljubljana.

Morgan, T. (2002) Business Rules and Information Systems: aligning IT with Business Goals. Addison Wesley. Chapter Defining Business, 2002

Mota, S. A. (2005) Bases de Datos Activas.

Perez, Boggiano (2007) Aplicación para Reglas de NEgocio..Universidad Central de Las Villas. Bumble-Bee (2000) Parser Generator 1.12 Ed.

Zimbrão, G. Enforcement of Business Rules in Relational Databases Using Constraints. Birgit D. Heinrich H.,Sten Loecher. OCL as Specification Language for Business rules in Database Applications.

Autorización y Renuncia

Los autores authorizan a LACCEI para publicar el escrito en los procedimientos de la conferencia. LACCEI o los editors no son responsables ni por el contenido ni por las implicaciones de lo que esta expresado en el escrito

Authorization and Disclaimer

Authors authorize LACCEI to publish the paper in the conference proceedings. Neither LACCEI nor the editors are responsible either for the content or for the implications of what is expressed in the paper.