

UNIVERSIDAD CENTRAL “MARTA ABREU” DE LAS VILLAS

FACULTAD MATEMÁTICA, FÍSICA Y COMPUTACIÓN

LICENCIATURA EN CIENCIAS DE LA COMPUTACIÓN



Trabajo de Diploma

Tablas de decisión para el análisis de consistencia y chequeos de integridad de un conjunto de reglas de negocio expresadas en SBVR

Autor: Victor Marrero Shimko

Tutora: M. Sc. María Elena Martínez Busto

Curso Académico: 2012-2013

Dictamen

Hago constar que el presente trabajo fue realizado en la Universidad Central Marta Abreu de Las Villas como parte de la culminación de los estudios de la especialidad de Ciencia de la Computación, autorizando a que el mismo sea utilizado por la institución, para los fines que estime conveniente, tanto de forma parcial como total y que además no podrá ser presentado en eventos ni publicado sin la autorización de la Universidad.

Firma del autor

Los abajo firmantes, certificamos que el presente trabajo ha sido realizado según acuerdos de la dirección de nuestro centro y el mismo cumple con los requisitos que debe tener un trabajo de esta envergadura referido a la temática señalada.

Firma del tutor

Firma del jefe del Seminario

Pensamiento

Son vanas y están plagadas de errores las ciencias que no han nacido del experimento, madre de toda certidumbre.

Leonardo Da Vinci

Dedicatoria

A mis padres por guiarme en todo el camino

A mis abuelos por su cariño

A mi hermano por su confianza

A mi novia Evelyn por ser mi vida

Agradecimientos

A mi familia por su preocupación

A mi tutora por todo su apoyo

A mis compañeros de investigación por toda su ayuda

A mis amistades por estar ahí

A aquellos que hicieron posible la realización de este trabajo

Resumen

El enfoque de reglas de negocio está dirigido al desarrollo de aplicaciones más flexibles y adaptables ante los cambios que se producen en las políticas del negocio. Repetidas veces, estos cambios percibidos sobre las reglas que rigen el negocio, se producen sin llevar a cabo un análisis consciente de las afectaciones que pueden acarrear. SBVR (acrónimo de *Semantics of Business Vocabulary and Business Rules*) es un estándar que sirve de guía para la representación de reglas y vocabularios de negocio. Las reglas definidas mediante SBVR son consistentes con el vocabulario sin embargo pueden reflejar anomalías respecto a otras reglas. La investigación tiene como objetivo describir patrones semánticos de reglas modeladas en SBVR que pueden ser mapeados a reglas de producción o reglas de restricción de tal manera que se puedan aplicar ciertos procesos de validación y verificación, para este proceso se define un algoritmo basado en tablas de decisión que incluye la detección de anomalías provocadas por los cuantificadores que actúan sobre las condiciones de las reglas.

Como resultado se obtiene una herramienta que dado un conjunto de reglas de producción y restricción en formato XML, analiza su consistencia mediante el algoritmo implementado.

Abstract

The business rules approach is aimed at developing more flexible and adaptable applications to changes that occur in business policies. Repeatedly, these perceived changes on the rules governing the business, are produced without carrying out an analysis aware of the effects that may ensue. SBVR (Semantics of Business Vocabulary and Business Rules) is a standard that provides guidance for the representation of business rules and vocabularies. Defined by SBVR rules are consistent with the vocabulary but may reflect anomalies over other rules. The research aims to describe semantic patterns modeled in SBVR rules can be mapped to production rules or restriction rules so that they can apply certain validation and verification processes, for this process is defined an algorithm based on tables decision that includes the detection of anomalies caused by quantifiers that act on the conditions of the rules.

The result is a tool that given a set of production and restriction rules in XML format, analyzes consistent with the implemented algorithm.

Contenido

Introducción.....	1
CAPÍTULO 1. VALIDACIÓN SEMÁNTICA DE REGLAS DE NEGOCIO.....	7
1.1. Enfoque de reglas de negocio	7
1.1.1. Definición de regla de negocio	8
1.1.2. Modelo de hechos	9
1.1.3. Clasificación de reglas de negocio.....	10
1.2. Estándar para la representación de reglas y vocabularios de negocio	12
1.2.1. Conceptos claves del enfoque SBVR	14
1.2.2. Descripción del meta-modelo de SBVR.....	16
1.2.3. Operadores lógicos usados en SBVR	18
1.3. Métodos de validación y verificación de reglas de negocio	20
1.3.1. Lógica estructurada.....	22
1.3.2. Español estructurado.....	22
1.3.3. Diagramas de transición de estado.....	24
1.3.4. Árboles de decisión.....	28
1.3.5. Tablas de Decisión.....	30
1.4. Conclusiones parciales del capítulo	34
CAPÍTULO 2. TRANSFORMACIONES LÓGICAS APLICADAS A REGLAS DE NEGOCIO.....	37
2.1. Patrones tratables mediante transformaciones lógicas	37
2.1.1. Descripción general del problema	37
2.1.2. Patrones de reglas identificados.....	39
2.2. Descripción del algoritmo de validación basado en tablas de decisión	46
2.2.1. Construcción de la tabla de decisión.....	46
2.2.2. Reglas dependiente e independientes	51
2.2.3. Algoritmo de validación de conflictos e inconsistencias.....	52
2.2.4. Descripción del algoritmo a través de un ejemplo.....	56
2.3. Conclusiones parciales del capítulo	59
CAPÍTULO 3. HERRAMIENTA PARA LA VALIDACIÓN DE REGLAS DE NEGOCIO.....	61
3.1. Descripción del caso de estudio	61

3.2. Descripción de la herramienta.....	63
3.2.1. Documento de entrada	63
3.2.2. Ambiente de la herramienta.....	65
3.3. Prueba de la herramienta sobre el caso de estudio	66
3.3.1. Preparación del documento de entrada	66
3.3.2. Ejecución y resultados	67
3.4. Conclusiones parciales del capítulo	72
Conclusiones.....	73
Recomendaciones.....	74
Referencias Bibliográficas.....	75
Anexo 1: Reglas obtenidas para el área de nefrología.....	78

Índice de figuras

Figura 1-1 Representación gráfica abstracta de un hecho (Moreno Montes de Oca, 2008; Martínez Busto et al., 2010b).	10
Figura 1-2 Pirámide de Términos, Hechos y Reglas (Chappel, 2005; Moreno Montes de Oca et al., 2008; Martínez Busto et al., 2010).	10
Figura 1-3 Clasificación de reglas de negocio (Martínez Fernández, 2010).	12
Figura 1-4 SBVR en el marco de la arquitectura MDA (OMG, 2008).	14
Figura 1-5 Principales elementos del meta-modelo de SBVR (OMG, 2008).	18
Figura 1-6 Estructura de una tabla de decisión.	30
Figura 3-1 Ambiente de trabajo de la herramienta Validación de reglas de negocio.	65
Figura 3-2 Cargar documento XML.	68
Figura 3-3 Tabla de decisión para reglas asociadas a trasplante.	68
Figura 3-4 Validación de reglas de restricción.	69
Figura 3-5 Detectar contradicciones.	69
Figura 3-6 Eliminación de redundancias y solapamiento de reglas.	70
Figura 3-7 Detectar redundancias entre cuantificadores.	70
Figura 3-8 Exportar reglas a documento XML.	70
Figura 3-9 Tabla de decisión para reglas asociadas a remitidos a consulta.	71
Figura 3-10 Eliminación de redundancias y solapamiento de reglas.	71

Índice de tablas

Tabla 1-1 Cuantificadores de la lógica de predicados.	19
Tabla 1-2 Operadores lógicos proposicionales.	19
Tabla 1-3 Operadores modales.	20
Tabla 2-1 Ejemplo de tabla de decisión.	38
Tabla 2-2 Ejemplo de tabla de decisión.	39
Tabla 2-3 Cantidad de posibilidades.	47
Tabla 2-4 Estructura de una tabla de condiciones.	48
Tabla 2-5 Condiciones y acciones obtenidas de las reglas.	49
Tabla 2-6 Tabla de decisión obtenida.	50
Tabla 2-7 Ejemplo para reglas puras y mixtas.	51
Tabla 2-8 Ejemplo para reglas dependientes e independientes.	52
Tabla 2-9 Tabla de decisión obtenida a partir del proceso de construcción.	57
Tabla 2-10 Reglas puras obtenidas.	58
Tabla 2-11 Eliminación de redundancias y solapamiento de reglas.	58

Introducción

La identificación de todas las regulaciones asociadas a una organización o negocio dado y su posterior utilización en sistemas que las hagan cumplir es una de las principales líneas del proceso de desarrollo de software en la actualidad ya que posibilitan la separación de las normas o restricciones del sistema que las debe hacer cumplir. Las regulaciones en los negocios actuales evolucionan de una manera dinámica produciéndose en estas cambios o modificaciones, lo que trae como consecuencia muchas veces el desuso de los sistemas informáticos por parte de los especialistas del negocio o la implementación de nuevos sistemas que cumplan con las nuevas condiciones. Las reglas de negocio¹ (RN) constituyen un enfoque que ha ido ganando fuerza, fundamentalmente porque pueden ser usadas como un instrumento para desarrollar aplicaciones flexibles y bases de datos modificables con facilidad (Bajec et al., 2001).

Las reglas de negocio son definiciones explícitas que regulan cómo opera un determinado negocio y cómo el mismo es estructurado (Barnes et al., 1997; Youdeowei, 1997; Bajec et al., 2000; Date, 2000; Bajec et al., 2001) entiéndase por negocio cualquier tipo de servicio que ofrezca determinada institución. Las reglas de negocio describen las políticas, normas, operaciones, definiciones y restricciones presentes en una organización siendo de vital importancia para alcanzar los objetivos de esta. Las organizaciones funcionan siguiendo múltiples reglas, explícitas o tácitas, que están embebidas en procesos, aplicaciones informáticas, documentos, entre otros. En las aplicaciones clásicas de bases de datos estas reglas han estado dispersas dentro de la lógica de las aplicaciones, lo que ha hecho difícil su modelación y mantenimiento, si bien por el contrario se facilita su aplicación. En el enfoque de reglas de negocio los especialistas tienen la posibilidad de definir las reglas centralmente previo al desarrollo de las aplicaciones, de manera que las reglas se vuelven independientes de las aplicaciones que las usan a la vez que pueden ser reutilizadas, modificadas y administradas, facilitándose así su mantenimiento y

¹ A partir de este momento se usa indistintamente “reglas de negocio” y “reglas”.

escalabilidad; o sea, si cambia una regulación solo hay que reflejarlo en la definición de la regla y no en los múltiples controles que hacen uso de la misma (Bajec et al., 2005), para ello se puede utilizar un motor de reglas de negocio. El motor de reglas de negocio es un sistema que se configura para dar servicio a las necesidades de negocio a través de la definición de objetos y reglas.

Las reglas se construyen tomando como base un vocabulario formado por las palabras y frases establecidas por la comunidad de usuarios de un negocio particular. La captación y modelación de las reglas es un proceso que debe llevarse a cabo tanto por los desarrolladores de software como por los conocedores del negocio (Bailey et al., 2005).

La forma de captar las reglas en cualquier entidad u organización debe de realizarse en un lenguaje cercano al natural, a la forma en que la comunidad de usuarios expresan sus regulaciones.

Un vocabulario de negocio recoge cada uno de los términos usados para definir las reglas, y sus significados. La Semántica del Vocabulario de Negocio y de las Reglas de Negocio (SBVR) permite describir cómo realizar la modelación del vocabulario y el conjunto de reglas de manera que estas puedan ser capturadas. Esto no es suficiente ya que las reglas pueden no reflejar ideas precisas o incluso pueden resultar contradictorias entre sí, por tanto se requiere que el conjunto de reglas creado pase por un proceso de validación y verificación donde se analice su consistencia.

La validación y verificación se puede realizar utilizando técnicas y mecanismos diversos en dependencia de las características del conjunto de reglas a chequear, entre ellos se encuentran: tablas de decisión, árboles de decisión e inglés o español estructurado (Goedertier et al., 2005).

En el Grupo de Bases de Datos del Centro de Estudios Informáticos existe una tendencia al desarrollo de trabajos dentro del enfoque de reglas de negocio varios de los cuales se consideran antecedentes al presente trabajo:

- ♦ Tesis de Maestría con título “Validación de la consistencia de un conjunto de reglas de negocio” (Núñez, 2012). En este trabajo se realiza un estudio sobre

formas de validar la consistencia de un conjunto de reglas y la presencia de la metodología Proteus que propone los pasos para todo el ciclo de vida de las reglas de negocio. Se analizan diferentes métodos de validación, son ellos: árboles de decisión, español estructurado y tablas de decisión. Como resultado del análisis realizado se elige el método basado en tablas de decisión considerando las ventajas que ofrece. Se confecciona un algoritmo que valida la consistencia de un conjunto de reglas basado en tablas de decisión el cual se aplica en el área de nefrología, y se comprueba que detecta determinados tipos de anomalías.

- ♦ Trabajo de Diploma con título “Validación de consistencia basada en la lógica formal de reglas de negocio” (Santos, 2012). En este trabajo se realiza una evaluación de la metodología Proteus para la identificación, captura y validación de las reglas de negocio. Se proponen vías de validación de consistencia y análisis de conflictos para las reglas almacenadas en el repositorio representado mediante el empleo del SBVR.

La validación y verificación constituyen procesos de gran significación pues son utilizados para evaluar la calidad del conjunto de reglas de negocio, detectando o eliminando los distintos problemas que puedan presentar.

Los sistemas manejadores de reglas de negocio además de posibilitar la edición y modificación de reglas deben permitir su validación y verificación de manera que estas puedan ser sometidas a cambios sin que se viole su consistencia e integridad. De lo anterior queda definido el ***problema de investigación***:

Un problema recurrente en los procesadores de reglas de negocio es la omisión o menoscabo de los procesos de validación y verificación, de manera que se elimina o se realiza muy empíricamente el análisis de la consistencia y chequeos de integridad del conjunto de reglas, lo que produce que se viole la fiabilidad del sistema y el conjunto de reglas se vuelve inutilizable. Las reglas representadas en términos de SBVR si bien son sintácticamente correctas, y semánticamente consistentes con el vocabulario, pueden presentar entre sí inconsistencias y ambigüedades, por lo que se hace necesario encontrar una vía para su validación.

Para darle solución al problema de investigación se planteó el siguiente **objetivo general de investigación** que consiste en: Aplicar tablas de decisión para el análisis de consistencia y chequeos de integridad de un conjunto de reglas de negocio expresadas mediante SBVR.

Este queda desglosado en los siguientes **objetivos específicos**:

1. Identificar las anomalías que se pueden detectar mediante tablas de decisión.
2. Describir patrones de reglas expresadas en SBVR que pueden ser representadas mediante tablas de decisión.
3. Proponer transformaciones para expresar el conjunto de reglas de negocio desde SBVR a tablas de decisión.
4. Proponer un algoritmo que realice el análisis de consistencia y chequeos de integridad del conjunto de reglas que cumple con los patrones definidos.
5. Obtener la herramienta que realiza el análisis de consistencia y chequeos de integridad usando tablas de decisión sobre un conjunto de reglas de negocio expresadas en SBVR.
6. Validar la calidad de la herramienta sobre un conjunto de reglas para un caso de estudio.

Las **preguntas de investigación** planteadas son:

¿A qué tipo de reglas de negocio expresadas mediante SBVR se les puede realizar el análisis de la consistencia y chequeos de integridad utilizando tablas de decisión?

¿Es posible describir mediante un algoritmo el análisis de consistencia y chequeos de integridad de un conjunto de reglas que cumple con determinados patrones definidos sobre SBVR?

¿De qué forma puede ser transformado un conjunto de reglas de negocio expresadas en SBVR a tablas de decisión?

Justificación

Las reglas de negocio son imprescindibles para el funcionamiento de cualquier empresa o institución, así como para el sistema de información (SI) que da soporte a

sus procesos. El cambio constante de las reglas que rigen los negocios actuales, hace que se necesiten herramientas más completas, que incluyan los procesos de validación y verificación de manera que puedan ser eliminadas las anomalías que se presenten, complementando los procesos de captura y edición de las reglas. Actualmente una de las principales líneas de investigación es la utilización del estándar SBVR para modelar las reglas y el vocabulario, este omite la detección de anomalías entre pares de reglas.

La tesis está estructurada en tres capítulos. En el Capítulo 1 se aborda el enfoque de reglas de negocio, posteriormente se caracteriza SBVR, especificando los operadores lógicos que se utilizan para modelar las reglas y luego se analizan mecanismos de validación existentes prestando mayor atención a las tablas de decisión. En el Capítulo 2 se describe un conjunto de patrones semánticos de reglas que pueden ser modelados por SBVR así como su posible transformación a tablas de decisión y se propone un algoritmo para el análisis de consistencia y chequeos de integridad de un conjunto de reglas de negocio usando tablas de decisión. El Capítulo 3 está dedicado al trabajo con la herramienta, su diseño e implementación, así como la validación mediante su aplicación sobre un caso de estudio.

La tesis cuenta con un anexo en el que se especifica el conjunto de reglas obtenidas del análisis del caso de estudio.

CAPÍTULO 1.

VALIDACIÓN SEMÁNTICA DE REGLAS DE NEGOCIO

CAPÍTULO 1. VALIDACIÓN SEMÁNTICA DE REGLAS DE NEGOCIO

Las reglas de negocio componen un nuevo paradigma en el desarrollo de Sistemas de Información (SI). Los Sistemas de Gestión de Reglas de Negocio (SGRN) presentan dos ventajas fundamentales: la cercanía a los expertos del negocio y la reducción del tiempo de modificación de la lógica del negocio. Los SGRN se caracterizan por permitir el uso de un lenguaje cercano al natural, facilitando la redacción e interpretación de las reglas por parte de los expertos de negocio. Además, el uso de reglas de alto nivel permitiría actualizar y desplegar aplicaciones más rápidamente y el proceso de implementación de estas reglas es prácticamente automático puesto que se elimina la necesidad de modificar y recompilar el código fuente de las aplicaciones (Martínez Fernández, 2010). En este capítulo se aborda el enfoque de reglas de negocio especificando algunas de sus principales características y formas de clasificación. Posteriormente se aborda algunos de los elementos fundamentales de SBVR y se describen los operadores lógicos que utiliza en la representación de las reglas. Finalmente se abordan algunos de los principales mecanismos de validación, específicamente tablas de decisión.

1.1. Enfoque de reglas de negocio

Las reglas de negocio han estado siempre presentes en los diferentes negocios, disueltas ya sea en documentos, aplicaciones o en la mente de las personas vinculadas al negocio. El enfoque de reglas de negocio está dirigido a proporcionar a las personas del negocio un control directo sobre el funcionamiento del mismo. Para el profesional de la computación constituye una vía para el desarrollo de aplicaciones más completas con un mayor rango de vida puesto que las reglas son centralmente definidas separadas de las aplicaciones que las usan, garantizando una mayor flexibilidad y adaptabilidad en el desarrollo de las herramientas.

En el enfoque de reglas de negocio es necesario que tanto los expertos del negocio como los desarrolladores hablen en el mismo idioma. Las reglas deben ser captadas de forma automática en los SI, y separadas de la lógica de las aplicaciones para facilitar su posterior mantenimiento (Barnes et al., 1997; Youdeowei, 1997; Bajec et al., 2000; Date, 2000; Bajec, 2001; Bajec et al., 2001). De esta forma la máquina debe

hacer cumplir las reglas, controlando y conduciendo los procesos de negocio de acuerdo a sus estándares, políticas y procedimientos (Struck et al., 1999; Goedertier et al., 2006).

1.1.1. Definición de regla de negocio

Desde el surgimiento del término los autores abordan la definición de formas diferentes (Appleton, 1984; Rosca et al., 1995; Youdeowei, 1997; Hay et al., 1997; Ceri et al., 1997; BRG, 2000; Bajec et al., 2000; Morgan, 2002; Ross, 2003, 2010). Ross (2003, 2010), Bajec (2000) y otros autores se acogen a lo planteado en (Hay et al., 1997), definición que se asume como válida en el presente trabajo.

“Una regla de negocio es una sentencia que define o restringe algunos aspectos del negocio, establece restricciones a la estructura del negocio, controlando o influyendo en el comportamiento del mismo. No podrá ser fraccionada o descompuesta en reglas más detalladas, en caso de ser reducida perdería información importante sobre el negocio (Hay et al., 1997).”

De esta definición puede inferirse que las reglas de negocio deben cumplir con ciertas propiedades. Las reglas deben ser:

- Atómicas: no pueden ser descompuestas sin que pierdan información importante.
- No ambiguas: tienen solamente una obvia interpretación.
- Compactas: típicamente son sentencias cortas.
- Compatibles: aplican los mismos términos que son usados en el modelo de negocios, están orientadas al negocio.

La utilidad de las reglas de negocio está dada por las potencialidades que estas ofrecen. En general brindan las siguientes ventajas (Castelo, 2011):

- Separación de las reglas del código, lo que posibilita que los sistemas sean flexibles y de fácil mantenimiento.
- Modelamiento de lógica de negocio compleja, lo que posibilita la automatización en la toma de decisiones claves.

- Administración centralizada de reglas de negocio, lo que posibilita la implementación consistente de políticas corporativas y regulaciones gubernamentales.
- Expertos de negocio pueden buscar, cambiar y probar reglas de negocio críticas, lo que posibilita la visibilidad, transparencia y transacciones sin errores.

1.1.2. Modelo de hechos

Para la administración de las reglas se hace necesario definir estructuras que permitan el modelado de las mismas. Mediante estas estructuras se facilita la utilización de las reglas. El analista toma el conjunto de formas estructurales, predefinidas como base y a partir de este genera la representación textual equivalente de las reglas, que no necesariamente se corresponde con la vista externa, la cual deberá ser cercana al lenguaje del negocio.

El modelo de hechos se conforma por conceptos básicos y sus conexiones (Ross, 2003). Cada concepto posee una definición clara y precisa desde la perspectiva del negocio y debe ser reflejada en el glosario de términos, o sea, la colección de todos los términos y sus definiciones. La idea se basa en que todas las personas del negocio hablen en un mismo idioma de manera que no puedan existir ambivalencias.

Cada concepto básico del negocio debe incluirse como un término independiente en el modelo de hechos, para lo que debe satisfacer las condiciones siguientes:

- Básico: no se puede derivar de otros términos.
- Atómico: no puede ser dividido.
- Cognoscible: representa cosas que existen, conocimiento acerca del negocio.

Las conexiones que relacionan los conceptos son conocidas como hechos, estas poseen una forma estándar. Según Morgan un hecho es “una interrelación entre términos identificables en el modelo de hechos”, y que esta interrelación puede estar limitada por otros elementos descriptivos en orden de “especificar la aplicabilidad de la regla con precisión” (Morgan, 2002). En general, los hechos representan conocimiento esencial del negocio, relacionan los términos y se expresan mediante

sentencias, se expresan en lenguaje común y extienden el vocabulario del negocio. Todo hecho sigue una rigurosa estructura sujeto1-verbo-sujeto2, ver Figura 1-1.

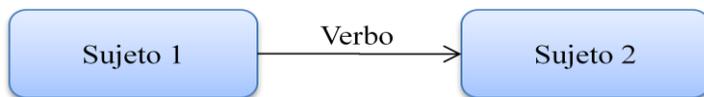


Figura 1-1 Representación gráfica abstracta de un hecho (Moreno Montes de Oca, 2008; Martínez Busto et al., 2010b).

El modelo de hechos trata las relaciones entre conceptos, además proporciona un esquema para organizar los demás componentes del negocio y sus atributos (Morgan, 2002). En la Figura 1-2 se muestra la relación que existe entre los términos, hechos y reglas en el modelo.

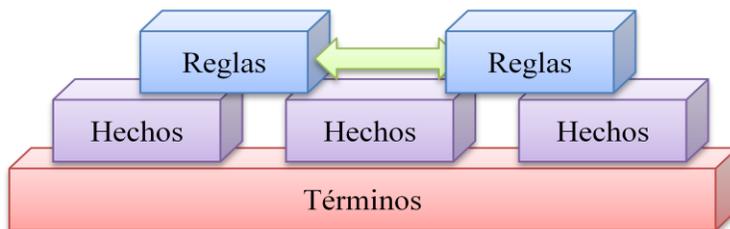


Figura 1-2 Pirámide de Términos, Hechos y Reglas (Chappel, 2005; Moreno Montes de Oca et al., 2008; Martínez Busto et al., 2010).

El modelo de hechos se enfoca en la estandarización de la terminología del negocio, para luego establecer un vocabulario común del negocio, por lo que se considera un punto de partida crucial en el modelado (Ross, 2000).

1.1.3. Clasificación de reglas de negocio

La clasificación de las reglas facilita el descubrimiento, análisis y finalmente la modelación de las mismas, por lo que es una tarea recomendada si bien no es obligatoria. Generalmente las reglas de negocio son descritas mediante un lenguaje de reglas formal o semi-formal, considerando que un lenguaje natural no es idóneo para este nivel. Por tanto, de acuerdo con las clasificaciones de las reglas, se pueden definir patrones asociados a cada categoría facilitando la labor. Existen varias taxonomías de clasificación según el nivel de abstracción, funcionalidad, finalidad, entre otros criterios, algunos de los cuales propuestos por autores como (Weiden et al., 2004; Ross, 2010). En el presente trabajo se abordan dos de estas taxonomías.

1.1.3.1. Clasificación desde la perspectiva de los expertos del negocio

Ross en (Ross, 2010) clasifica las reglas de negocio en reglas operativas y reglas estructurales desde la perspectiva de los expertos y analistas del negocio. Ross define como:

- Reglas operativas: aquellas que pueden ser violadas por algún agente del negocio. Por ejemplo, la regla “todo auto de renta debe poseer una matrícula con el código de la región” es una regla operativa que puede ser quebrantada en un momento dado. Si una regla de este tipo es violada entonces se producirá una penalización. Para la regla que nos ocupa pudiera ser que el auto deje de ser un auto de renta.
- Reglas estructurales: aquellas que permiten definir cómo se organiza o estructura el negocio y que nunca pueden ser violadas de forma directa. Definen características que permiten categorizar las entidades que intervienen en el negocio. Por ejemplo, la regla “un cliente es distinguido si ha realizado más de 3 rentas de autos en el último año” es una regla estructural y no puede ser violada directamente. La aplicación errónea de esta regla conduce a resultados erróneos o incoherentes y no a sanciones como el caso de una regla operativa. Las reglas de inferencia y de cálculo se consideran estructurales.

Esta taxonomía es muy importante ya que es la que aborda SBVR como se verá más adelante.

1.1.3.2. Clasificación desde la perspectiva del negocio

Desde el punto de vista del negocio una regla puede provenir de distintas expresiones tales como, una restricción, una pauta, un cálculo, una inferencia o una acción (Martínez Fernández, 2010). En la Figura 1-3 se muestra ejemplos de distintas expresiones que representan los distintos tipos de reglas de negocio. Esta taxonomía puede ser considerada como cercana a la implementación ya que establece patrones que las reglas deben simular, y por tanto guía el proceso de obtención de las reglas en el modelo de negocio.

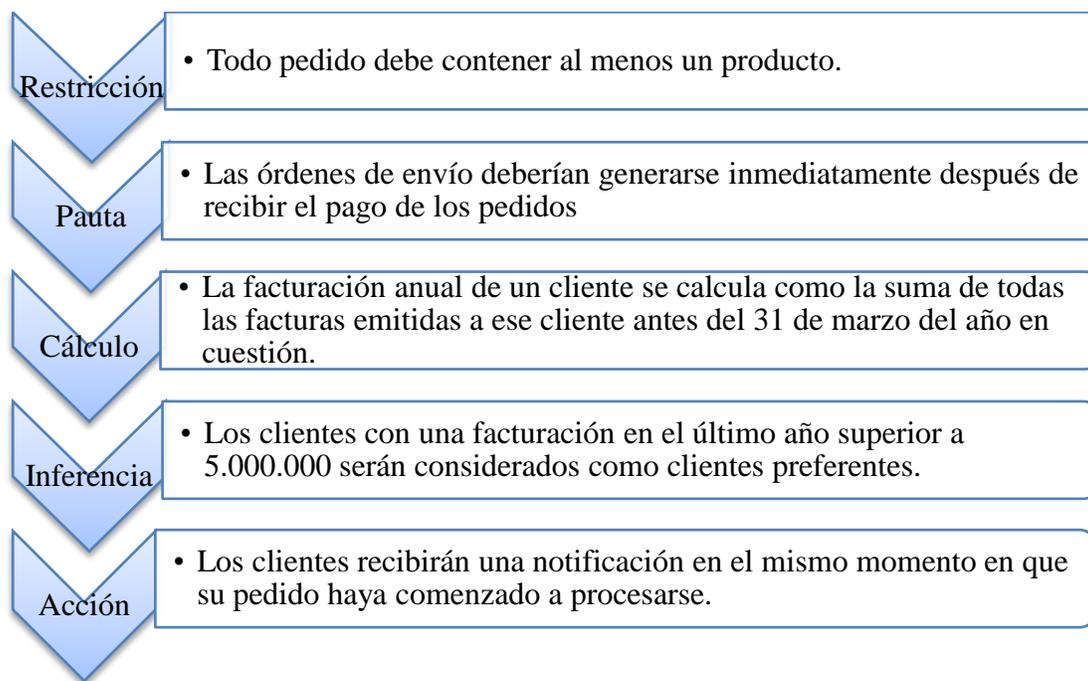


Figura 1-3 Clasificación de reglas de negocio (Martínez Fernández, 2010).

1.2. Estándar para la representación de reglas y vocabularios de negocio

En los modernos procesos de negocio los analistas determinan y administran las reglas, los sistemas de administración de reglas de negocio son empleados para separar la lógica del negocio del código de la aplicación. Separar la lógica tiene como principal fin que las aplicaciones posean una mayor adaptabilidad, de manera que pueda extenderse su vida útil ante los cambios que se producen en la política de las empresas que le dan soporte a dichas aplicaciones. SBVR provee una forma de documentar la semántica del vocabulario y las reglas especificados en un lenguaje familiar a los usuarios del negocio. Es independiente al lenguaje que se utilice para especificar dichas reglas y vocabularios. SBVR pretende lograr un acercamiento entre el lenguaje que esgrimen los expertos para hablar del negocio y las tecnologías de la información (IT) que le dan soporte (Vanthienen, 2008).

El lenguaje SBVR ha sido definido por el grupo dedicado al estudio de las reglas de negocio llamado Equipo de Reglas de Negocio (del inglés *Business Rule Team* - BRT). Es una respuesta a la solicitud de propuestas (del inglés *Request For Proposal* - RFP) realizado por el Grupo de Gestión de Objetos (OMG acrónimo del inglés

Object Management Group) y denominada Semántica de Negocio para Reglas de Negocio (BSBR acrónimo del inglés *Business Semantics for Business Rules*) (OMG, 2003).

SBVR es un estándar adoptado por la OMG, que se propuso ser la base para una descripción declarativa, formal y detallada de la lengua natural de una entidad compleja. Este estándar es considerado un vocabulario, o un conjunto de sub-vocabularios, formados por una serie de entradas terminológicas. Cada una de estas entradas incluye una definición, junto con otras especificaciones como notas y ejemplos. A menudo, las entradas también incluyen reglas (necesidades) para el elemento en concreto que se está definiendo.

Está diseñado para ajustarse a la capa de modelo de negocio de la Arquitectura Dirigida por Modelos (MDA acrónimo del inglés *Model Driven Architecture*), ver Figura 1-4, y representa especificaciones de lógica formal expresadas en lenguaje natural controlado, para ser procesadas por medios automáticos. Si bien el vocabulario y las reglas SBVR se ubican en la capa de componente del modelo del negocio (CIM) de MDA, pueden ser mapeados a las capas MDA-PIM y MDA-PSM, que no son incluidas como parte del estándar. Esta ubicación en la arquitectura tiene dos implicaciones:

- Se centra en reglas y en vocabularios de negocio, especialmente en aquellos relevantes para ser utilizados junto con las reglas.
- Los modelos de negocio, incluyendo aquellos que soporta SBVR, describen el negocio y no los sistemas de información que los soportan.

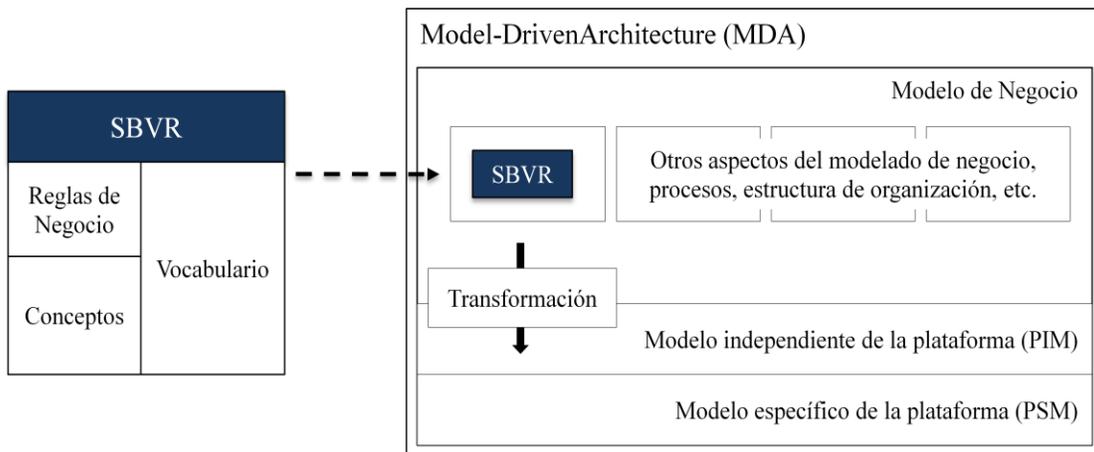


Figura 1-4 SBVR en el marco de la arquitectura MDA (OMG, 2008).

SBVR constituye una recopilación del estado del arte en los diferentes campos de conocimiento relacionados con el enfoque de reglas de negocio, establece un vocabulario consistente, definido de forma tal que se satisfacen los requerimientos planteados en el Manifiesto de Reglas de Negocio (del inglés: *Business Rules Manifesto*) (BRG, 2003).

1.2.1. Conceptos claves del enfoque SBVR

A continuación se tratan algunos de los conceptos más importantes que maneja SBVR:

La semántica

La semántica se define como “*el significado o la relación de los significados de un signo o conjunto de signos*”. En SBVR, los signos no se limitan al texto, ya que pueden ser de cualquier forma: palabras, frases, códigos, números, iconos, sonidos, entre otros.

A su vez, los significados se definen como “*aquello que significa una palabra, un signo, una sentencia o una descripción, aquello que alguien intenta expresar o algo que se puede entender*”, lo que se puede comprender como “*algo percibirle o concebible*”.

SBVR incluye dos vocabularios especializados:

- El “Vocabulario para la descripción del vocabulario del negocio”, se ocupa de todos los tipos de términos y significados.

- El “Vocabulario para la descripción de las reglas de negocio”, que se ocupa de la especificación del significado de las reglas de negocio y se construye a partir del anterior.

Los dos han sido separados para que el Vocabulario para la descripción de vocabularios de negocios pueda utilizarse de manera independiente.

Vocabulario del negocio

Un vocabulario de negocios contiene todos los términos especializados, nombres, y tipos de hechos, formas de conceptos que una organización o comunidad de negocio utilizan.

El vocabulario para la descripción de vocabularios de negocios propuesto por SBVR se basa en varios estándares terminológicos que han sido utilizados durante décadas para relacionar vocabularios multilingües.

Reglas y Lógica Formal

Un aspecto esencial en el tratamiento que SBVR da al concepto de "regla" es su consistencia con la lógica formal. Según (OMG, 2008) expertos notables en este ámbito recomiendan que el mejor tratamiento para la interpretación de las reglas debiera incluir restricciones de necesidad y obligatoriedad. En consecuencia, en SBVR, una regla es, “un elemento de orientación que introduce una obligación o una necesidad”, las dos categorías fundamentales de reglas descritas en el apartado 1.1.3.1, reglas estructurales (que expresan necesidades) y reglas operativas (que expresan obligaciones).

Reglas, Tipos de Hechos y Conceptos expresados a través de términos

De manera informal, un tipo de hecho es una asociación entre dos o más conceptos, como por ejemplo “Auto de Renta está localizado en Filial”.

En SBVR, las reglas siempre se construyen mediante la aplicación de necesidad u obligación de los tipos de hechos. Por ejemplo, la regla de "un alquiler no debe tener más de tres choferes adicionales" se basa en el tipo de hecho "Alquiler tiene Chofer Adicional". De esta forma, SBVR asume un principio básico del enfoque de reglas de

negocio, o sea “Las reglas de negocio se construyen sobre tipos de hechos, y los tipos de hechos se construyen a partir de los conceptos expresados por los términos”.

Una consecuencia importante del enfoque de SBVR es que los conceptos (incluidos los tipos de hechos) son distintos de las reglas. Este diseño de SBVR permite dar soporte a los conceptos (incluidos los tipos de hechos) para que puedan ser empleados, opcionalmente, para la construcción de vocabularios empresariales.

Todas las reglas de negocio deben ser procesables, es decir, aptas para ser validadas, lo que significa que si un persona que conozca la regla de negocio dada una situación puede decidir directamente si el negocio está cumpliendo o no con la regla. Que las reglas de negocio sean procesables no significa que sean automatizables. Muchas reglas de negocio, especialmente operativas, no pueden ser automatizables por los sistemas de información, por tanto deben ser implementadas como actividades de usuario u otros procedimientos manuales.

Intercambio Semántico

El meta-modelo SBVR está diseñado para proporcionar interfaces de datos estandarizados e intercambios de datos estandarizados entre herramientas que recogen, organizan, analizan y utilizan reglas y vocabularios de negocio, así como herramientas que relacionan vocabularios y reglas de negocio a otros modelos e implementaciones.

Una característica importante del meta-modelo SBVR es la forma en que ha sido creado. Comienza con los Vocabularios SBVR acompañado de un conjunto de reglas que permite transformar el vocabulario a MOF (*Meta-data Object Facility*) / XMI (*XML Meta-data Interchange*), lenguajes estándar para la representación de modelos.

1.2.2. Descripción del meta-modelo de SBVR

SBVR define un meta-modelo para reglas y vocabularios de negocios orientados a la descripción del negocio en vez de orientarlo a una implementación técnica, dando un vocabulario a los expertos del dominio para que puedan expresar reglas de negocio (Amaolo, 2011). Por ello, el meta-modelo se puede analizar a través de sus aspectos básicos indicados en la Figura 1-5: la comunidad, el cuerpo de conocimiento

(significados) compartido, la formulación semántica, la lógica formal y la representación del negocio. Estos son abordados a continuación.

Comunidad

La base del vocabulario del negocio es la comunidad. En el nivel de negocio la comunidad de principal importancia son las empresas para las que se están estableciendo y expresando las reglas. Sin embargo, otras comunidades como: la industria en la que opera una empresa, grupos de estandarización, autoridades reguladoras, entre otras, también necesitan ser reconocidas.

Hay que destacar que las sub-comunidades pueden necesitar acceso a los significados compartidos que pueden estar recogidos en diferentes vocabularios, desde jerga especializada a correspondencias en diferentes idiomas.

Significados compartidos

Una comunidad tiene un conjunto de significados compartidos que comprende conceptos y reglas de negocio. Lo que se comparte es el significado, no la forma de expresarlo. Para poder intercambiar, discutir y validar significados compartidos deben haber sido expresados previamente de alguna forma. La estructura del conjunto de significados compartidos se define a través de la asociación de conceptos abstractos, tipos de hecho y reglas de negocio, y no asociando sentencias en un lenguaje determinado.

Formulación lógica

La formulación lógica proporciona una sintaxis formal, abstracta e independiente del lenguaje para la captura de la semántica de un conjunto de significados compartidos. Esta soporta múltiples formas de representación, tales como: nombres y formas verbales.

La formulación lógica soporta dos características esenciales de SBVR. La primera es la asociación de un conjunto de significados compartidos a vocabularios usados por las comunidades. La segunda la correspondencia con el estándar XMI, que permite el intercambio de conceptos, hecho y reglas entre herramientas que soportan SBVR.

Representación de negocio

Los conceptos y reglas de negocio en un conjunto de significados compartidos necesitan ser representados en los vocabularios admitidos por las comunidades que comparten dichos significados. Estos vocabularios pueden expresarse en diferentes idiomas, en lenguajes artificiales como UML o en subconjuntos especializados de determinados idiomas.

Lógica formal

SBVR tiene un fundamento teórico sólido, basado en la lógica formal, que permite tanto la formulación lógica como las estructuras de conjuntos de significados compartidos. La base es la lógica de predicados de primer orden.

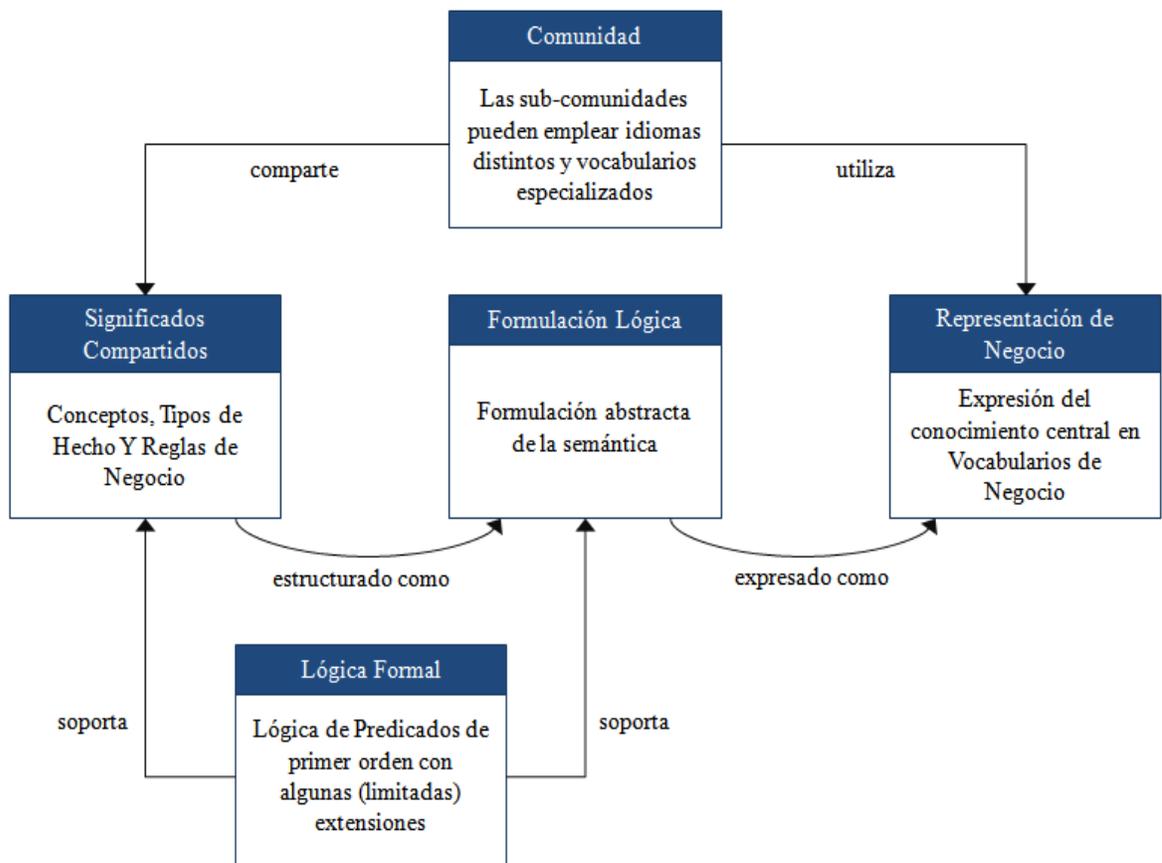


Figura 1-5 Principales elementos del meta-modelo de SBVR (OMG, 2008).

1.2.3. Operadores lógicos usados en SBVR

Como se ha mencionado anteriormente SBVR posee un fuerte basamento teórico sobre la lógica de predicados de primer orden con algunas extensiones a la lógica modal, esto lo hace coherente con lo planteado en el artículo 5 del Manifiesto de

Reglas de Negocio (BRG, 2003): “Las lógicas formales, como la lógica de predicados, son fundamentales para la expresión formal de reglas en términos de negocio, así como para las tecnologías que implementan dichas reglas.”

A continuación se muestran los cuantificadores y operadores modales o proposicionales de los que hace uso. Las letras ‘ n ’ y ‘ m ’ representan números enteros. Las letras ‘ p ’ y ‘ q ’ representan expresiones de proposiciones.

Tabla 1-1 Cuantificadores de la lógica de predicados.

Cada	cuantificación universal
Algún	cuantificación existencial
al menos uno	cuantificación existencial
al menos n	cuantificación al-menos- n
a lo sumo uno	cuantificación a-lo-sumo-uno
a lo sumo n	cuantificación al-lo-sumo- n
exactamente uno	cuantificación exactamente-uno
exactamente n	cuantificación exactamente- n
al menos n y a lo sumo m	cuantificación de rango numérico
más que uno	cuantificación al-menos- n con $n=2$

Tabla 1-2 Operadores lógicos proposicionales.

no es el caso que p	negación lógica
p y q	conjunción
p o q	disyunción
p o q pero no ambos	disyunción exclusiva
si p entonces q	Implicación
q si p	Implicación

p si y solo si q	Equivalencia
no ambos p y q	formulación nand
ni p ni q	formulación nor
p tanto o no q	formulación tanto-o-no

Este último operador está compuesto por dos operandos lógicos, o sea dos proposiciones, donde ‘ p ’ se considera un consecuente y ‘ q ’ un inconsecuente. O sea, el valor lógico de ‘ q ’ no altera el resultado de aplicar el operador lógico, mientras que el valor de ‘ p ’ se obtiene como resultado de aplicar el operador lógico. En general los operadores de implicación poseen al menos un antecedente (premisa) y al menos un consecuente (conclusión).

Cuando un sujeto se repite al usar ‘y’ o ‘o’ el sujeto repetido puede elidirse. Por ejemplo, la declaración “Un conductor tiene una licencia y un conductor respeta el tráfico”, puede abreviarse a: “Un conductor tiene una licencia y respeta el tráfico”. De manera similar, pueden elidirse un sujeto y un verbo repetidos. Por ejemplo, la declaración, “Un conductor tiene una licencia y un conductor tiene un carro”, puede abreviarse a: “Un conductor tiene una licencia y un carro”.

Tabla 1-3 Operadores modales.

es obligatorio que p	formulación de obligación	Operativos
es prohibido que p	formulación de obligación dentro de una negación lógica	
es permitido que p	formulación de permisibilidad	
es necesario que p	formulación de necesidad	Estructurales
es imposible que p	formulación de necesidad dentro de una negación lógica	
es posible que p	formulación de posibilidad	

1.3. Métodos de validación y verificación de reglas de negocio

La verificación es el proceso dirigido a la detección de inconsistencia, incompletitud o redundancia en un conjunto de reglas de negocio sin considerar el significado de las

reglas, esto significa que no le importa acerca de si la regla es correcta o no (MEJIA, 2011).

Aun si se puede probar que las reglas son lógicamente consistentes y completas, las reglas podrían llevar a resultados incorrectos pero lo harán en una manera consistente.

Validación es el proceso que apunta por la detección de resultados incorrectos o comportamiento indeseado (MEJIA, 2011).

La forma más simple de validar reglas es pasar las reglas cambiadas a otro miembro de la organización. En un contexto de tecnologías de información, la validación frecuentemente se hace probando la aplicación y verificando los resultados, o comparando los resultados con previos resultados que se creen están correctos.

Algunos de los términos usados para señalar problemas en un conjunto de reglas se muestran a continuación (Castelo, 2011):

- **Inconsistencia:** Es una condición en un conjunto de reglas de negocio que ocurre cuando 2 o más reglas llevan a comportamientos o resultados conflictivos.
- **Incompletitud:** Es una condición en un conjunto de reglas que ocurre cuando existe un caso de negocio que lleva a un resultado o comportamiento indefinido.
- **Redundancia:** Es una condición en un conjunto de reglas de negocio que ocurre cuando existe una regla de negocio que no tiene contribución significativa al posible comportamiento o posibles resultados.
- **Anomalía:** Es un término más general para inconsistencia, incompletitud o redundancia.

Un conjunto de reglas posee alta calidad cuando no se producen las situaciones que se muestran a continuación (Ross, 2009):

- Una regla es similar a otra regla.
- Una regla subsume otra regla.
- Una regla es lógicamente equivalente a otra regla.
- Una regla está en conflicto con otra regla.

Existen diferentes métodos para la modelación de reglas de negocio los cuales aplican diferentes algoritmos para la validación y verificación de reglas de negocio entre los cuales podemos citar la lógica estructurada, los diagramas de transición de estado (redes de Petri), las tablas de decisión y los árboles de decisión.

1.3.1. Lógica estructurada

Mediante este enfoque se realiza la modelación lógica de las reglas del negocio utilizando las diferentes construcciones de la programación estructurada.

- si...entonces
- caso1...caso2...caso3...o entonces
- en cuanto...
- hasta que...
- de 1 a n

A partir de la lógica estructurada se modela y describe la ejecución de acciones y se definen procedimientos de cálculo y de decisión.

1.3.2. Español estructurado

El español estructurado es otro método para evitar los problemas de ambigüedad del lenguaje al establecer condiciones y acciones, tanto en procedimientos como en decisiones. Este método no hace uso de árboles o tablas; en su lugar utiliza declaraciones para describir el proceso. El proceso no muestra reglas de decisión; las declara (Landinez, 2009).

Aun con esta característica, las especificaciones en español estructurado requieren que el analista primero identifique las condiciones que se presentan en un proceso y las decisiones que se deben tomar cuando esto sucede junto con las acciones correspondientes. Sin embargo, el método también permite hacer una lista de todos los pasos en el orden que se llevan a cabo. Para ello no se utilizan símbolos y formatos especiales, características de los árboles y las tablas de decisión que para algunos resultan incómodos. Además, es posible describir con rapidez los procedimientos en su totalidad ya que para ello se emplean declaraciones muy similares al español.

La terminología utilizada en la descripción estructurada de una aplicación consiste, en gran medida, en nombres de datos para los elementos que están definidos en el diccionario de datos desarrollado para el proyecto.

El español estructurado emplea tres tipos básicos de declaraciones para describir un proceso: estructuras de secuencia, estructuras de decisión y estructuras de iteración.

Estructuras de secuencia: Una estructura de secuencia es un solo paso o acción incluido en un proceso. Este no depende de la existencia de ninguna condición y cuando se encuentra siempre se lleva a cabo. En general, se emplean varias instrucciones en secuencia para describir un proceso.

Estructuras de decisión: El español estructurado es otro camino para mostrar el análisis de decisión. Por tanto, a menudo se incluyen las secuencias de acciones dentro de estructuras de decisión que sirven para identificar condiciones. Es así como las estructuras de decisión aparecen cuando se pueden emprender dos o más acciones, lo que depende del valor de una condición específica. Para esto primero se evalúa la condición y después se toma la decisión de emprender las acciones o el grupo de acciones asociados con esta condición. Una vez determinada la condición las acciones son incondicionales.

Estructuras de iteración: En las actividades rutinarias de operación, es común encontrar que algunas de ellas se repiten mientras existen ciertas condiciones o hasta que estas se presentan. Las instrucciones de iteración permiten al analista describir estos casos.

Como puede observarse, el español estructurado puede ser de utilidad para describir con claridad condiciones y acciones. Cuando se examina el ambiente de una empresa, los analistas pueden usar el español estructurado para declarar las reglas de decisión que se aplican en este medio. Si los analistas no pueden declarar que acción emprender cuando se toma una decisión, entonces necesitan adquirir mayor información para descubrir la situación. Por otro lado, después de describir las actividades en forma estructurada, los analistas pueden pedir a otras personas que revisen la descripción y determinen con rapidez los errores u omisiones cometidos al establecer los procesos de decisión.

1.3.3. Diagramas de transición de estado

Un diagrama de transición de estado o red de Petri (RP) es un grafo dirigido que utiliza dos tipos de nodos: localizaciones y transiciones. Las localizaciones se representan mediante círculos y las transiciones con rectángulos. Los nodos se conectan entre sí mediante arcos dirigidos y no están permitidas las conexiones entre nodos de un mismo tipo (Van Der Aalst, 1998)

Una definición más rigurosa es:

Una RP es una tripleta (P, T, F) , donde:

- P es un conjunto finito de localizaciones.
- T es un conjunto finito de transiciones.
- $F \subseteq (P \times T) \cup (T \times P)$ es un conjunto de arcos.

Las redes de Petri pueden ser utilizadas para modelar reglas de negocio. Las propiedades estructurales de las RP permiten a su vez realizar el análisis de validación y verificación del conjunto de reglas que ha sido representado por la red.

Esta metodología exige que las reglas se expresen como enunciados de la lógica formal. Posteriormente estos enunciados condicionales deben ser transformados en una representación equivalente en RP. La transformación se basa en una asignación entre las operaciones lógicas de conjunción (\wedge), disyunción (\vee), y la implicación y las nociones de sincronización, concurrencia, elección, etc., de la teoría de RP. Una vez que una base de reglas se transforma en una RP, la solución al problema se convierte en una aplicación directa de herramientas de análisis de la teoría de RP. La validación y verificación de la base de reglas se realiza primero mediante la explotación de las propiedades estructurales de la representación de RP y luego mediante la construcción del gráfico de ocurrencias directamente a partir de la representación de RP (Zaidi et al., 1995).

Hay que señalar que un par de conceptos mutuamente exclusivos no pueden ser identificados a menos que hayan sido definidos.

Anomalías que se pueden detectar mediante RP

- Reglas redundantes

Ejemplo 1:

$$(p_1 \wedge p_2 \rightarrow A)$$

$$(p_2 \wedge p_1 \rightarrow A)$$

Ejemplo 2:

$$(q_1 \wedge q_2 \wedge q_3 \wedge q_4 \rightarrow A)$$

$$(q_1 \wedge q_2 \rightarrow p_1)$$

$$(q_3 \wedge q_4 \rightarrow p_2)$$

$$(p_1 \wedge p_2 \rightarrow A)$$

- Reglas subsumidas

Ejemplo:

$$(p_1 \rightarrow A)$$

$$(p_1 \wedge p_2 \rightarrow A)$$

- Reglas inconsistentes

- Contradicción directa: En este caso μ es un conjunto de predicados mutuamente excluyentes.

Ejemplo 1:

$$(\alpha \rightarrow \beta) \text{ where } \alpha, \beta \in \mu$$

Ejemplo 2:

$$(q_1 \wedge q_2 \rightarrow p_1)$$

$$(q_3 \wedge q_4 \rightarrow p_2)$$

$$(p_1 \wedge p_2 \rightarrow \neg q_1)$$

- Contradicción en la conclusión: En este caso se presentan antecedentes iguales que implican conclusiones mutuamente excluyentes.

Ejemplo 1:

$$(\alpha \rightarrow \beta)$$

$$(\alpha \rightarrow \tau) \text{ where } \beta, \tau \in \mu$$

Ejemplo 2:

$$(p_1 \wedge p_2 \rightarrow p_3)$$

$$(p_3 \wedge p_4 \rightarrow A)$$

$$(p_1 \wedge p_2 \wedge p_4 \rightarrow \neg A)$$

- Contradicción en las premisas

Ejemplo 1:

$$(\alpha \wedge \beta \rightarrow \tau) \text{ where } \alpha, \beta \in \mu$$

Ejemplo 2:

$$(p_1 \wedge p_2 \rightarrow p_3)$$

$$(p_3 \wedge \neg p_1 \rightarrow A)$$

- Reglas Circulares

Ejemplo 1:

$$(p_1 \rightarrow p_2)$$

$$(p_2 \rightarrow p_3)$$

$$(p_3 \rightarrow p_1)$$

Ejemplo 2:

$$(p_1 \rightarrow p_2 \wedge p_3)$$

$$(p_2 \wedge p_4 \rightarrow p_5)$$

$$(p_3 \wedge p_5 \rightarrow p_4)$$

- Incompletitud: Una base de conocimiento es incompleto cuando no tiene la información necesaria para responder a una pregunta (apropiadamente) de interés para el sistema.
 - Reglas con condiciones ambiguas: Presencia de al menos un concepto complejo en la premisa que no pueden ser explicados o definidos en términos de los conceptos básicos.
$$(A \wedge p_1 \rightarrow R)$$
 - Reglas con conclusiones no funcionales: Dada una entrada, si la conclusión derivada no es el concepto principal o todos o algunos de sus atributos no se puede concluir a partir de la base de reglas, el conjunto de normas aplicables a esta entrada es incompleto.
$$(p_1 \wedge p_2 \rightarrow R)$$
 - Reglas aisladas: Una regla es aislada si y solo si: a) Todas las proposiciones en su premisa son conceptos complejos que no se pueden explicar en términos de

los conceptos básicos y b) Su conclusión R no es ningún concepto principal y no existe regla que conduzca desde R hasta un concepto principal.

Una regla individual de la forma " $P_1 \wedge P_2 \wedge \dots \wedge P_n \rightarrow Q$ " se transforma en una RP con una sola transición con n localizaciones de entrada, cada una representada por una única proposición P_i , y una localización de salida que representa la afirmación. Las etiquetas de las localizaciones y transiciones corresponden a las proposiciones y las reglas a las que representan. La regla de inferencia es implementada por el mecanismo de ejecución de la teoría de RP. Un símbolo en una localización representa la asignación de verdad de una proposición. Si todas las localizaciones de la preselección de una transición que representa la premisa de una regla que se cumple se marcan, la regla (transición) está activada y se puede ejecutar (disparar) marcando todos los consecuentes (lugares de salida) como válidos (marcado). Todo el conjunto de reglas de decisión se obtiene mediante la unificación de todas las reglas individuales. El proceso representa la relación causal entre las reglas y los hechos de la base de conocimientos. Posteriormente las entradas básicas y conceptos principales definidos para la base de reglas se agrupan en las entradas y salidas virtuales, P_{in} y P_{out} , con transiciones T_{in} y T_{out} , respectivamente.

Mediante los análisis estáticos y dinámicos de la teoría de RP se pueden detectar las siguientes anomalías:

Análisis estático

- Reglas ambiguas
- Reglas no funcionales
- Reglas circulares
- Reglas inconsistentes

Análisis dinámico

- Reglas redundantes
- Reglas inconsistentes

La representación mediante RP tiene asociadas ventajas y desventajas algunas de las cuales se mencionan a continuación:

Ventajas

- Usan construcciones bien conocidas.
- Estas construcciones son muy intuitivas si el número de reglas no es muy grande.

Desventajas

- La complejidad crece fuera de control cuando la base de reglas es muy grande.
- Circularidades especiales y otros problemas no pueden tratarse.
- Funcionan para bases de conocimiento proposicionales.
- No contemplan incertidumbre ni control
- La "flecha" de las reglas es la implicación material, y no se contempla la idea de que pueda haber una lógica subyacente en la que la flecha represente "relaciones relevantes" entre antecedentes y consecuentes.
- Carecen de la semántica para expresar tiempo.

1.3.4. Árboles de decisión

Cuando un proceso de decisión estructurada se integra con ramificaciones complejas, entonces se hace uso de los árboles de decisiones. El árbol de decisión es un método para el análisis de decisiones que consiste de nodos y ramas. Se dibujan sobre un plano horizontal, con la raíz del árbol al lado izquierdo del papel y las ramas hacia la derecha. Esto permite al analista describir las condiciones de acciones sobre las ramas. Son más adecuados cuando se deben realizar acciones en una secuencia determinada (Soria et al., 2009; Martínez et al., 2009).

El árbol de decisión representa en forma secuencial condiciones y acciones, muestra qué condiciones se consideran en primer lugar, cuáles en segundo y así sucesivamente. Este método también permite mostrar la relación que existe entre cada condición y el grupo de acciones permisibles asociado con ella. Cada vez que se ejecuta un árbol de decisión solo un camino será seguido dependiendo del valor actual de la variable evaluada.

Los árboles de decisión son normalmente contruidos a partir de la descripción de la narrativa de un problema. Se recomienda el uso del árbol de decisión cuando el número de acciones es pequeño y no son posibles todas las combinaciones.

La raíz del árbol es el punto donde comienza la secuencia de decisión. La rama a seguir depende de las condiciones existentes y de la decisión que debe tomarse. Al avanzar de izquierda a derecha por una rama en particular se obtiene una serie de toma de decisiones. Después de cada punto de decisión se encuentra el siguiente conjunto de decisiones a considerar. De esta forma, los nodos del árbol representan condiciones y señalan la necesidad de tomar una determinación relacionada con la existencia de alguna de estas, antes de seleccionar la siguiente trayectoria. A la derecha del árbol se encuentran las acciones que deben realizarse, las que a su vez dependen de la secuencia de condiciones que las proceden.

El árbol de decisión tiene tres ventajas principales sobre las tablas de decisión, y se muestran a continuación:

- Toma las ventajas de la estructura consecutiva de las ramas del árbol de decisiones, de tal forma que se identifican de manera inmediata el orden de verificación de las condiciones y las acciones que se deben llevar a cabo.
- Las condiciones y acciones del árbol de decisiones se encuentran en ciertas ramas pero no en otras, a diferencia de la tabla de decisiones, donde todas forman parte de la misma tabla.
- Al compararse con las tablas, los árboles de decisiones se entienden con más facilidad en una organización y son más apropiados como un método de comunicación.

Los árboles de decisión no siempre son la mejor herramienta para el análisis de decisiones. El árbol de decisión de un sistema complejo con muchas secuencias de pasos y combinaciones de condiciones puede tener un tamaño considerable. El gran número de ramas que pertenecen a varias trayectorias constituye más un problema que una ayuda para el análisis. En estos casos el analista corre el riesgo de no

determinar qué políticas o estrategias de la empresa son la guía para la toma de decisiones específicas.

1.3.5. Tablas de decisión

La tabla de decisión es una herramienta que sintetiza procesos en los cuales se dan un conjunto de condiciones y un conjunto de acciones a tomar según el valor que toman las condiciones (Castilla, 2007). Puede utilizarse como instrumento en los distintos momentos de un proyecto, o sea: en la exposición de los hechos, en el análisis del sistema actual, en el diseño del nuevo sistema y en el desarrollo del software.

Estructura de una tabla de decisión

La tabla de decisión está integrada por: matriz de condiciones, matriz de acciones, y matriz de reglas para condiciones y acciones como se muestra en Figura 1-6. En la matriz de condiciones se enumeran todas las situaciones que pueden presentarse. Las reglas de condiciones indican qué valor debe asociarse a cada una de las condiciones. En la matriz de acciones se enlistan el conjunto de todos los pasos que se deben seguir cuando se presentan ciertas condiciones. Las reglas de acciones muestran las acciones específicas del conjunto que deben emprenderse dados los valores que toman las condiciones.



Figura 1-6 Estructura de una tabla de decisión.

Procedimiento para llenar la tabla

El proceso de llenado de la tabla requiere de la aplicación de los siguientes pasos:

- A. Identificar cuáles son las condiciones y cuáles las acciones. Las condiciones son una serie de circunstancias que pueden darse o no y que conducen a desarrollar

ciertas actividades que son las acciones. Cada condición puede tomar uno de los siguientes tres valores: si se cumple (S), no se cumple (N), o no interesa su valor (-). Las acciones solo pueden tomar dos valores: se realiza (X), o no se realiza (-).

- B. Extraer un listado de las condiciones y de las acciones.
- C. Completar la tabla en las matrices de condiciones y acciones, teniendo en cuenta las siguientes consideraciones:
 - a. Si hay condiciones entre sí y acciones entre sí, expresadas de distinta manera pero que significan lo mismo, deben colocarse una sola vez, enunciadas en forma sencilla y sintética, normalizando el lenguaje.
 - b. Si hay condiciones que son opuestas, por ejemplo: varón, mujer o mayorista, minorista, debe colocarse una de ellas porque por la negativa se obtendrá la otra.
 - c. Para hacer más simple el enunciado de las condiciones pueden utilizarse, donde sea conveniente, operadores relacionales (>, <, =, <>, etc.).
- D. Identificadas las condiciones y las acciones se colocan en sus respectivas matrices y se construyen después las reglas. Las reglas se asocian a las columnas de la matriz.

Tipos de reglas

A las reglas que en la parte de condiciones poseen únicamente entradas S y/o N se las llama reglas puras; a las reglas que en la parte de condiciones poseen por lo menos una entrada indiferente (-), se las llama reglas mixtas. De aquí se puede inferir que cada regla mixta se puede descomponer en dos o más reglas puras.

Tipos de anomalías identificables

A través de tablas de decisión pueden ser identificadas diversos tipos de anomalías distinguiéndose entre casos de redundancia, ambivalencia, circularidad y deficiencia. De acuerdo al lugar donde ocurren se pueden distinguir dos tipos de anomalías: intra-tabular e inter-tabular. La anomalía intra-tabular ocurre entre los componentes dentro de una misma tabla, sin embargo la inter-tabular ocurre durante la interacción entre los componentes de diferentes tablas. Consecuentemente con esto la relación entre las tablas debe ser verificada.

- Redundancia
 - Par de columnas duplicadas: Cuando dos reglas iguales desde el punto de vista semántico son introducidas en la tabla.
 - Par de columnas subsumidas: Se presenta cuando dos reglas son parecidas, es decir, realizan las mismas acciones, donde los valores de las condiciones de una regla son un caso particular de las de la otra.
 - Fila de condición redundante o irrelevante: Es cuando se presenta una condición irrelevante entre dos reglas, es decir, la condición no influye en las acciones a realizar.
 - Fila de acción inusual: Se presenta cuando entre dos reglas un estado de una condición nunca se satisface.
- Ambivalencia
 - Ambivalencia a nivel de columna: Son dos reglas que en la tabla de decisión se representan en las columnas, bajo los mismos valores de condiciones realizan acciones contradictorias.
 - Ambivalencia a nivel de fila: Dos o más filas en la tabla de decisión tienen entradas contradictorias en una misma columna, es decir en una misma regla.
- Circularidad: Cuando se presenta un ciclo entre diferentes tablas de decisión.
- Deficiencia
 - Reglas aisladas: Es cuando una regla hace referencia a una tabla no usada.
 - Columnas vacías: Cuando en una regla no hay entradas de acciones a realizar.
 - Acciones no referenciadas: Cuando se hace referencia a estados de condiciones o acciones que nunca se satisfacen o nunca se ejecutan.

Depuración de las tablas de decisiones

Tiene como objetivo hacer más compacta la tabla, es decir que tenga menos reglas y encontrar posibles errores en las reglas por haber partido, por ejemplo, de un enunciado contradictorio.

Primera ley: Unificación de reglas por indiferencia

Se analizan todos los pares de reglas posibles, si se encuentra un par de reglas en las que se dé que tengan idénticas entradas de condiciones y acciones, excepto en una condición donde una registra una entrada S y la otra registra una entrada N, ambas pueden ser eliminadas de la tabla y reemplazadas por una regla con idénticas entradas de condiciones y acciones que registren la entrada indiferente (-) en aquella condición donde las originales diferían.

Criterio: Reglas independientes y dependientes

Una vez aplicada la primera ley deben contemplarse todos los pares de reglas posibles. Si en un par se observa que por lo menos para una condición una regla registra una entrada S y la otra una entrada N, se dice que ambas reglas son independientes, el par de reglas que no cumple con este requisito se dice dependiente. Si las reglas son dependientes es posible determinar si son redundantes o contradictorias.

Un par de reglas dependientes son redundantes cuando las reglas tienen las mismas entradas de acciones y contradictoria cuando tienen diferentes entradas de acciones. Si son redundantes puede aplicarse la segunda y tercera ley, sino significa que hay un error en el enunciado o que se ha construido mal la tabla por lo que hay que investigar el error antes de continuar.

Segunda ley: Eliminación de reglas redundantes

Si en un par de reglas dependientes y redundantes, una es pura y la otra es mixta, la pura está contenida en la mixta y debe ser eliminada.

Tercera ley: Eliminación de reglas redundantes

Si en un par de reglas dependientes y redundantes ambas son mixtas, existe al menos una regla pura común a ambas que puede eliminarse de una de ellas.

Encadenamiento de tablas

Al construir tablas de decisión podemos encontrar que:

- a) En varios procesos distintos existen condiciones y acciones que son comunes. Cuando esto ocurre podemos construir una tabla con las condiciones y acciones que se repiten en las distintas tablas y agregar en las tablas originales una acción

que ordene aplicar o ejecutar la tabla con las condiciones y acciones que son comunes.

- b) Se trate de un proceso muy complejo con muchas condiciones y acciones y que resulte conveniente desdoblar la tabla en dos más.

En ambas situaciones puede ocurrir que la transferencia entre las distintas tablas sea definitiva o temporal.

Ventajas de la utilización de tablas de decisión

- Brindan una poderosa presentación compacta y estructurada.
- Previenen errores en las reglas muy fácilmente, tales como, inconsistencias, redundancias, contradicciones y situaciones imposibles.
- Organiza el conocimiento de una forma modular.
- Promueve la integridad y precisión en el análisis de decisiones estructuradas.
- Ayuda a la toma de decisiones.

Las tablas de decisión han sido estudiadas o aplicadas en muchos contextos de investigación debido a las facilidades que presentan. Un enfoque que se maneja actualmente es el uso de tablas de decisión para la gestión de reglas de negocio. La técnica de tablas de decisión está siendo continuamente usada en la representación y la validación de la lógica de negocios complejos a través del modelamiento de las reglas. Las tablas de decisión demuestran seguir siendo usadas en los SI que dan soporte a las empresas y específicamente en los SGRN. De ahí que en la presente investigación se haga uso de las tablas de decisión como método para realizar la validación semántica de las reglas de negocio.

1.4. Conclusiones parciales del capítulo

En este capítulo se especificó la situación del estado del arte en las líneas de investigación en que se centra el trabajo. La exposición se divide en tres fragmentos principales. El primer fragmento aborda el enfoque de reglas de negocio, en este se aborda la definición de regla de negocio de Hay (1997), considerada muy acertada, luego se abordan algunos conceptos fundamentales del enfoque tales como hecho y modelo de hechos y por último se exponen dos perspectivas de clasificación de reglas

de negocio las cuales se consideran importantes para la investigación. En el segundo fragmento se abordan algunos conceptos fundamentales de SBVR, se da una descripción informal del meta-modelo de SBVR y se presentan los operadores lógicos que utiliza SBVR para conformar las reglas. En el tercer fragmento se abordan algunos de los principales métodos de validación y verificación de reglas de negocio haciendo énfasis en las tablas de decisión como el más acertado. Luego de todo este despliegue se llegan a las siguientes conclusiones:

1. Se aborda la clasificación de las reglas en operativas y estructurales tomada de (Ross, 2010) la cual es la misma que utiliza SBVR.
2. Se establece el estándar SBVR como forma de representación de vocabularios y reglas de negocio fuertemente basados en la lógica formal.
3. Se identifican varios métodos de validación centrandose el trabajo en tablas de decisión y detallando las anomalías que pueden detectarse.

CAPÍTULO 2.

TRANSFORMACIONES LÓGICAS APLICADAS A REGLAS DE NEGOCIO

CAPÍTULO 2. TRANSFORMACIONES LÓGICAS APLICADAS A REGLAS DE NEGOCIO

Una regla representada en términos de SBVR puede ser procesada para luego ser incorporada en un repositorio de reglas. Las reglas capturadas basándose en SBVR son sintácticamente correctas, y semánticamente consistentes con el vocabulario que se define mediante SBVR. Sin embargo, las reglas entre sí pueden ser inconsistentes o ambiguas, pudiendo ser depuradas mediante otros procesos de validación. En este capítulo se especifica un conjunto de patrones de reglas soportadas por SBVR que pueden ser representados mediante tablas de decisión definiendo cómo deben ser transformadas las reglas sin que estas pierdan su significado. Luego se describe el algoritmo de validación cuyas bases se apoyan en las tablas de decisión. En este apartado se manejan entre otros, ejemplos del caso de estudio Trasplante Renal. Finalmente se concluye sobre la utilidad del algoritmo y la descripción de los patrones.

2.1. Patrones tratables mediante transformaciones lógicas

Una transformación no es más que un cambio que se produce en una regla de SBVR de manera que la regla resultante pueda ser representada mediante tablas de decisión u otro método de validación, sin que se pierda el significado lógico de la regla. La transformación del conjunto de reglas permite que puedan ser aplicados los procesos de análisis de la consistencia que ofrecen los métodos de validación. Esto hace que se puedan detectar anomalías entre pares de reglas, lo cual no se garantiza con la sola utilización de SBVR, salvo mediante el análisis humano exhaustivo.

2.1.1. Descripción general del problema

En este sub-epígrafe se describe a través de un ejemplo la necesidad de apoyarse en métodos de validación que complementen la labor que desempeña SBVR para lograr una mayor calidad en el conjunto de reglas de negocio.

A continuación se muestran dos reglas que pueden ser descritas mediante SBVR partiendo de un vocabulario definido para el caso de estudio Trasplante Renal. Ambas reglas por separado son sintácticamente correctas y semánticamente

consistentes con el vocabulario, sin embargo si se analizan juntas se observan contradicciones.

Regla 1: Es necesario que todo paciente que presente insuficiencia renal crónica y filtrado glomerular inferior a 15 ml/min sea remitido a consulta de hemodiálisis.

Regla 2: Es necesario que todo paciente que presente insuficiencia renal crónica sea remitido a consulta de nefrología.

La regla 1 plantea que es necesario remitir a consulta de hemodiálisis aquellos pacientes que cumplen con las dos condiciones propuestas, mientras que para la regla 2 se plantea que es necesario remitir a consulta de nefrología aquellos pacientes que cumplen con la primera condición de la regla 1. Como en la regla 2 no interesa que el filtrado glomerular sea inferior a 15 ml/min entonces un paciente que cumpla con las condiciones de la regla 1 también deberá ser remitido a consulta de nefrología por la regla 2, o sea que se presenta la contradicción: ¿a cuál consulta debe ser remitido el paciente?

Este tipo de anomalías no se detecta al utilizar SBVR por lo que requieren un tratamiento especial. En ese sentido la utilización de las tablas de decisión muestra su utilidad. La tabla de decisión asociada al ejemplo anterior se muestra a continuación:

Tabla 2-1 Ejemplo de tabla de decisión.

Condiciones	R1	R2
paciente presenta insuficiencia renal crónica	S	S
paciente presenta filtrado glomerular inferior a 15 ml/min	S	-
Acciones		
paciente es remitido a consulta de nefrología	X	-
paciente es remitido a consulta de hemodiálisis	-	X

Las reglas 1 y 2 son dependientes contradictorias por lo que existe al menos una regla pura común a ambas que debe ser eliminada. Esta eliminación debe ser realizada por parte del especialista o conocedor del negocio, pues en última instancia es quien determina cual regla es la que está mal escrita. Si las reglas presentaran las mismas

acciones entonces serían dependientes redundantes y la regla 1 se podría eliminar directamente.

La regla 2 es una regla mixta que puede ser descompuesta en dos reglas puras como se observa en la siguiente tabla, en donde las reglas R1 y R2a son contradictorias, o sea una de ellas incumple con los términos del negocio.

Tabla 2-2 Ejemplo de tabla de decisión.

Condiciones	R1	R2a	R2b
paciente presenta insuficiencia renal crónica	S	S	S
paciente presenta filtrado glomerular inferior a 15 ml/min	S	S	N
Acciones			
paciente es remitido a consulta de nefrología	X	-	-
paciente es remitido a consulta de hemodiálisis	-	X	X

Mediante el ejemplo descrito puede observarse la importancia de las tablas de decisión así como el “por qué” de su uso en la presente investigación.

2.1.2. Patrones de reglas identificados

En este apartado se describen algunos patrones de reglas en SBVR que pueden ser modelados mediante tablas de decisión con el objetivo de poder realizar validaciones semánticas sobre este conjunto de reglas. Hay que señalar que la captura de estos patrones no se implementa, solamente se describen. La herramienta encargada de la validación asume que los patrones de reglas ya han sido capturados.

No existe un estándar de cómo hacer reglas de negocio, pero si es posible hacer algunas recomendaciones. Los patrones pueden reflejar la forma de las cosas, los hechos que en la industria o los tipos de problemas que un sistema automatizado pretende negociar (Morgan, 2002). Permite describir las reglas que pertenecen a una característica particular de acuerdo a su semántica.

Los patrones de reglas son recursos semánticos usados para describir las reglas de negocio. Las reglas restringen la semántica de los conceptos y de las relaciones

conceptuales de un dominio de aplicación particular. Los patrones representan la sintaxis permitiendo describir la estructura de las reglas. La sintaxis de dichos patrones se define utilizando conceptos del vocabulario del negocio manejados en el modelo de hechos (Witt, 2009a, b, c).

Con el apoyo semántico que brinda el modelo de hechos de SBVR, se definen un conjunto de patrones que sirven como estructura para la descripción de las reglas.

Entre las estructuras que se utilizan podemos citar las condiciones, los cuantificadores y las acciones.

<cuantificador> representa los cuantificadores descritos por SBVR (ver Tabla 1-1) a excepción del cuantificador universal. De manera más formal puede definirse como sigue:

<cuantificador> ::= (exactamente | al menos | a lo sumo | al menos <cardinalidad> y a lo sumo) <cardinalidad>

<cardinalidad> ::= <entero positivo>

<acción> son descritas a través de los hechos. Las formas verbales conjugadas como infinitivos o participios suelen ser utilizadas para declarar acciones. Por ejemplo:

<acción> ::= [<sujeito-1>] ((se debe | se tiene que) <verbo infinitivo>) | ((debe ser | tiene que ser) <verbo participio>) (a | con | al) <sujeito-2>

<condición> se basa en un hecho que puede o no tener restricciones indicadas por el cuantificador, de la forma:

<condición> ::= <sujeito1> [no] <verbo> [cuantificador] <sujeito2>

Donde los corchetes indican opcionalidad. Por ejemplo:

Paciente tiene al menos tres turnos

Cuya negación sería

Paciente no tiene al menos tres turnos

o lo que es lo mismo:

Paciente tiene a lo sumo dos turnos

Las tablas de decisión están compuestas por reglas de producción que poseen la siguiente sintaxis general:

If <condición> **then** <acción>

<acción> **If** <condición>

Pueden existir múltiples condiciones, negadas o no, enlazadas por conjunciones. También pueden existir múltiples acciones unidas por conjunciones. En el caso de las acciones no se utiliza el operador unario de negación pues si la acción no se realiza, simplemente no aparece en la conclusión de la regla.

Conjunción:

De manera más específica entonces una regla se puede representar como:

If <cond-1> **and** ... **and** <cond-n> **then** <acc-1> **and** ... **and** <acc-m>

<acc-1> **and** ... **and** <acc-m> **If** <cond-1> **and** ... **and** <cond-n>

Disyunción:

En el caso de reglas con disyunciones que enlazan condiciones se les aplica la siguiente transformación para poderlas representar mediante las tablas de decisión.

Dada la regla:

If <cond-1> **or** ... **or** <cond-n> **then** <acc-1> **and** ... **and** <acc-m>

Se transforma en:

If <cond-1> **then** <acc-1> **and** ... **and** <acc-m>

...

If <cond-n> **then** <acc-1> **and** ... **and** <acc-m>

Para las acciones no se hace uso del operador lógico ‘or’ ya que para ciertas condiciones se realizará siempre un único conjunto de acciones.

Los operadores proposicionales que se utilizan a continuación pueden ser reducidos a fórmulas que solo presentan operadores de conjunción, disyunción o negación.

Disyunción exclusiva:

Dada la regla:

If <cond-1> xor <cond-2> then <acc-1> and ... and <acc-m>

Se transforma en:

If <cond-1> and not <cond-2> then <acc-1> and ... and <acc-m>

If not <cond-1> and <cond-2> then <acc-1> and ... and <acc-m>

Formulación nand:

Dada la regla:

If <cond-1> nand <cond-2> then <acc-1> and ... and <acc-m>

Se transforma en:

If not <cond-1> and not <cond-2> then <acc-1> and ... and <acc-m>

If <cond-1> and not <cond-2> then <acc-1> and ... and <acc-m>

If not <cond-1> and <cond-2> then <acc-1> and ... and <acc-m>

Formulación nor:

Dada la regla:

If <cond-1> nor <cond-2> then <acc-1> and ... and <acc-m>

Se transforma en:

If not <cond-1> and not <cond-2> then <acc-1> and ... and <acc-m>

Forma normal disyuntiva:

En general, dada una regla, pueden existir múltiples condiciones conectadas con operadores de la lógica proposicional que utiliza SBVR (ver Tabla 1-2). Si se asume que cada condición representa una proposición, entonces partiendo de las condiciones de la regla se puede hallar una forma normal disyuntiva equivalente (Barwise et al., 1999), como se muestra a continuación:

If (<cond-1> and ... and <cond-n>)₁ or ... or (<cond-1> and ... and <cond-n>)_k then <acc-1> and ... and <acc-m>

Donde en cada una de las sub-fórmulas separadas por las disyunciones, no necesariamente aparecen todas las condiciones. A partir de una regla cuyas condiciones están en forma normal disyuntiva se derivan las siguientes reglas:

If (<cond-1> **and**... **and** <cond-n>)₁ **then** <acc-1> **and** ... **and** <acc-m>

...

If (<cond-1> **and**... **and** <cond-n>)_k **then** <acc-1> **and** ... **and** <acc-m>

Para transformar la parte de las condiciones de una regla a forma normal disyuntiva deben seguirse las siguientes pautas.

- a) Verificar que no existan operadores de implicación o doble implicación.
- b) Expresar las operaciones de disyunción exclusiva, formulación nand y formulación nor en términos de conjunciones, disyunciones y negaciones.
- c) Aplicar las leyes de De Morgan y de la doble negación para eliminar todas las negaciones que no actúan sobre una única condición. $(\neg(a \vee b) \equiv \neg a \wedge \neg c)$
- d) Utilizar las leyes distributivas y asociativas hasta obtener la forma normal disyuntiva. $((a \vee b) \wedge c \equiv (a \wedge c) \vee (b \wedge c))$

Una tabla de decisión indica que en una regla, siempre que las condiciones se cumplan las acciones deberán ser ejecutadas. Por tanto si se está en presencia de una regla donde el consecuente se puede ejecutar o no, dicha regla no será compatible con la representación mediante tablas de decisión. Este es el caso de la regla operativa “es permitido que p” y la regla estructural “es posible que p” donde ‘p’ es una regla de producción. Seguidamente se presentan algunos patrones de reglas de SBVR que pueden ser representados o ser de utilidad, para la validación mediante tablas de decisión.

Patrón 1. Regla de producción operativa o estructural

Es obligatorio que p \equiv No es permitido que no p \equiv Es prohibido que no p

Es necesario que p \equiv No es posible que no p \equiv Es imposible que no p

Los patrones antes expuestos poseen el mismo significado lógico que la regla de producción ‘p’ ya sea de carácter necesario u obligatorio, o sea, si las condiciones son verdaderas las acciones deben ser ejecutadas. Por tanto su representación en una tabla de decisión coincide con la de la regla ‘p’.

En las reglas ‘No es obligatorio que p’, ‘No es obligatorio que no p’, ‘No es necesario que p’ y ‘No es necesario que no p’ se infiere que la acción puede no ejecutarse, a

pesar de que las condiciones se satisfagan, debido a lo que no son compatibles con las reglas de producción representables en la tabla de decisión.

En las reglas ‘Es obligatorio que no p’ y ‘Es necesario que no p’ niega la regla ‘p’ lo que hace que pierda sentido representarla en la tabla de decisiones puesto que la regla de producción ‘p’ no se cumple.

Patrón 2. Reglas de producción representadas mediante cuantificadores

En este patrón se muestra el uso de cuantificadores específicamente el ‘para todo’ o ‘todo’ (\forall) y su transformación a reglas de producción

(Es necesario que | Es obligatorio que) todo <suje-to-A0> que <verbo-0> [cuantificador-0] <suje-to-B0> [[[y [[que] <verbo-1>] [cuantificador-1] <suje-to-B1>]] (y [[que] <verbo-2>] [cuantificador-2] <suje-to-B2>)]... (y [[que] <verbo-n>] [cuantificador-n] <suje-to-Bn>)] ((debe ser | tiene que ser | sea) <verbo participio-1> (a | con | al) <suje-to-1>) y ... y ((debe ser | tiene que ser | sea) <verbo participio-m> (a | con | al) <suje-to-m>)

En este caso el ‘que’ se utiliza para introducir una restricción.

Ejemplo 1:

Es obligatorio que todo paciente que posea insuficiencia renal crónica sea enviado a consulta de nefrología.

Regla de producción equivalente:

Si paciente posee insuficiencia renal crónica **entonces** paciente es enviado a consulta de nefrología.

Ejemplo 2:

Es necesario que todo paciente que posea insuficiencia renal crónica o que posea bajo índice de masa muscular sea enviado a consulta de hemodiálisis.

Reglas de producción obtenidas:

Si paciente posee baja masa muscular **entonces** paciente es enviado a consulta de hemodiálisis.

Si paciente posee insuficiencia renal crónica **entonces** paciente es enviado a consulta de hemodiálisis.

Patrón 3. Reglas de restricción

(Es necesario que | Es obligatorio que) [todo | el | la] <suje-to-A0> <verbo-0> [cuantificador-0] <suje-to-B0> [[[y [[(todo | el | la) <suje-to-A1>] <verbo-1>] [cuantificador-1] <suje-to-B1>]...] [y [[(todo | el | la) <suje-to-An>] <verbo-n>] [cuantificador-n] <suje-to-Bn>]]

Las reglas de tipo restricción son almacenadas en una lista para verificar luego en las reglas almacenadas en la tabla de decisión si sus condiciones cumplen con las restricciones.

Es necesario que el hospital tenga al menos 3 médicos de guardia y cada paciente sea atendido por al menos un enfermero.

Elementos de la lista:

Hospital tiene al menos tres médicos de guardia

Paciente es atendido por al menos un enfermero

Patrón 4. Reglas de restricción

(Un | Una) <suje-to-A0> [no] debe <verbo-0> [cuantificador-0] <suje-to-B0> [[[y [[(un | una) <suje-to-A1>] [no] debe <verbo-1>] [cuantificador-1] <suje-to-B1>]...] [y [[(un | una) <suje-to-An>] [no] debe <verbo-n>] [cuantificador-n] <suje-to-Bn>]]

Ejemplo 1:

Un paciente debe tener al menos un médico asignado

Elementos de la lista:

Paciente tiene al menos un médico asignado

Dichos patrones permiten establecer un subconjunto de reglas estructuradas que pueden ser captadas mediante una herramienta de software. Como las reglas se escriben usando como guía SBVR entonces el lenguaje en que estas se declaran es cercano a los usuarios del negocio. Las herramientas de edición de reglas de negocio deben basar su trabajo en estructuras que permitan la manipulación segura de las

reglas en forma comprensible para los especialistas del negocio, sin inconsistencias ni ambigüedades.

2.2. Descripción del algoritmo de validación basado en tablas de decisión

En este epígrafe se describe el algoritmo de validación implementado basado en tablas de decisión el cual adiciona verificaciones sobre el uso de cuantificadores en las reglas.

Las reglas de producción capturadas conforman la tabla de decisión. Las reglas de restricción son almacenadas en una lista a partir de la cual se verifica que las condiciones en las reglas de producción cumplan con las restricciones.

Se parte de un fichero XML con un conjunto de reglas de producción y restricción editadas. El algoritmo abarca dos etapas fundamentales, la captura de las reglas y conformación de la tabla de decisión, y el proceso de validación y análisis de la consistencia.

La utilización de las tablas de decisión como mecanismo para complementar la validación de reglas modeladas en SBVR se debe a las facilidades que estas ofrecen (Ross, 2005a, b; Vanthienen, 2012)

2.2.1. Construcción de la tabla de decisión

Mediante el análisis del documento XML se obtienen cada una de las reglas. Las reglas de restricción son almacenadas en una lista que es utilizada luego en el proceso de validación. De cada regla de restricción se conocen los hechos que participan en la regla así como los cuantificadores (si existen) sobre cada uno de los hechos. De cada regla de producción se conoce si su comportamiento es admitido o no (se analiza más adelante) y para cada condición se conoce que hecho y que cuantificador (si existe) conforman la condición. Con el conjunto de reglas de producción se construye la tabla de decisión.

Como se explicó en el Capítulo 1 la tabla de decisión está integrada por: matriz de condiciones, matriz de acciones y matriz de reglas para condiciones y acciones. En la matriz de condiciones se enumeran todas las situaciones que puedan presentarse. Para cada regla ciertas acciones serán activadas (S|N). Por otro lado, en la matriz de

acciones se enlistan todas las posibles acciones, de manera tal que en cada regla ciertas acciones serán activadas. Seguidamente se detalla un método para construir dicha tabla de decisiones.

Para construir una tabla de decisión el analista necesita definir el tamaño máximo de la tabla. Los siguientes pasos conforman un método sistemático para el desarrollo de tablas de decisiones (Ross, 2006).

1. Determinar el número de condiciones que pueden afectar a las decisiones. Combine las filas que se solapan, por ejemplo, condiciones que se excluyen mutuamente (al menos 3, a lo sumo 2). El número de condiciones coincide con el número de filas del cuadrante superior derecho de la tabla.
2. Determinar el número de acciones posibles que pueden ser tomadas. Este número determina el número de filas del cuadrante inferior derecho de la tabla.
3. Determinar el número de alternativas para cada condición. En una tabla de entradas limitadas solo son posibles dos alternativas (Si-No) o (Verdadero-Falso) para cada condición. En una tabla de entradas extendidas por cada condición existen muchas alternativas y el procedimiento se hace muy engorroso.
4. Calcular el número máximo de columnas en la tabla de decisión, para ello se suman el número de alternativas de cada condición de un mismo hecho y se multiplica por el número de alternativas de otras condiciones basadas en otros tipos de hechos. Si hay cuatro condiciones donde cada una tiene dos alternativas (S y N) y las tres primeras se basan en un mismo hecho existirán doce posibilidades, según se indica en la Tabla 2-3.

Tabla 2-3 Cantidad de posibilidades.

Condición 1:		2 alternativas	*
Condición 2:	+	2 alternativas	
Condición 3:		2 alternativas	
Condición 4:	+	2 alternativas	
$(2+2+2)*2=12$ posibilidades			

5. Rellenar las alternativas de las distintas condiciones. Para esto se debe tener en cuenta que en una misma regla no se deben activar dos condiciones basadas en un mismo hecho. Una tabla de condiciones asociada a la tabla anterior sería:

Tabla 2-4 Estructura de una tabla de condiciones.

S	S	N	N								
				S	S	N	N				
								S	S	N	N
S	N	S	N	S	N	S	N	S	N	S	N

6. Completar la tabla insertando X en todas las acciones que debe ejecutarse con cada regla.

Una vez obtenido el conjunto de condiciones se debe verificar que si una condición basada en un tipo de hecho utiliza algún cuantificador entonces todas las condiciones basadas en ese mismo hecho deben utilizar también algún cuantificador. De esta manera si se tiene “<Sujeto1> <Verbo> <Sujeto2>” puede obtenerse “<Sujeto1> <Verbo> al menos 1 <Sujeto2>”.

Con el siguiente ejemplo se observa el proceso de construcción de la tabla de decisión.

Descripción del ejemplo

A partir de las siguientes reglas se obtiene la Tabla 2-5 con el conjunto de condiciones y acciones detectadas.

R1: Si un cliente pertenece a la empresa y el pago se produce en a lo sumo 30 días y la cantidad solicitada no supera la cantidad en stock entonces se debe facturar con descuento y enviar mercadería solicitada.

R2: Si un cliente pertenece a la empresa y el pago se produce en al menos 31 días y la cantidad solicitada no supera la cantidad en stock entonces se debe facturar sin descuento y enviar mercadería solicitada.

R3: Si un cliente no pertenece a la empresa y la cantidad solicitada no supera la cantidad en stock entonces se debe facturar sin descuento y enviar mercadería solicitada.

R4: Si un cliente pertenece a la empresa y el pago se produce en a lo sumo 30 días y la cantidad solicitada supera la cantidad en stock entonces se debe facturar con descuento y enviar cantidad en stock y dejar pendiente la cantidad solicitada menos el stock.

R5: Si un cliente no pertenece a la empresa y el pago se produce en a lo sumo 30 días y la cantidad solicitada supera la cantidad en stock entonces se debe facturar sin descuento y enviar cantidad en stock y dejar pendiente la cantidad solicitada menos el stock.

R6: Si el pago se produce en a lo sumo 20 días y la cantidad solicitada no supera la cantidad en stock entonces se debe facturar sin descuento y enviar mercadería solicitada y enviar premio de bonificación.

De las reglas anteriores se obtiene las siguientes condiciones y acciones.

Tabla 2-5 Condiciones y acciones obtenidas de las reglas.

Condiciones	Acciones
Cliente pertenece a la empresa	Facturar con descuento
Pago se produce en a lo sumo 30 días	Enviar mercadería solicitada
Pago se produce en al menos 31 días	Facturar sin descuento
Pago se produce en a lo sumo 20 días	Enviar cantidad en stock
Cantidad solicitada supera la cantidad en stock	Dejar pendiente la cantidad solicitada menos el stock
	Enviar premio de bonificación

Posteriormente se normalizan los cuantificadores y se construye la tabla de decisión. Durante el proceso de normalización de los cuantificadores para cada condición con

cuantificador ‘a lo sumo’ se encuentra una condición equivalente que utiliza el cuantificador ‘al menos’.

Tabla 2-6 Tabla de decisión obtenida.

Condiciones	Reglas					
	R1	R2	R3	R4	R5	R6
Cliente pertenece a la empresa	S	S	N	S	N	-
Pago se produce en al menos 31 días	N	S	-	N	S	
Pago se produce en al menos 21 días			-			N
Cantidad solicitada supera la cantidad en stock	N	N	N	S	S	N
Acciones						
Facturar con descuento	X	-	-	X	-	-
Enviar mercadería solicitada	X	X	X	-	-	X
Facturar sin descuento	-	X	X	-	X	X
Enviar cantidad en stock	-	-	-	X	X	-
Dejar pendiente la cantidad solicitada menos el stock	-	-	-	X	X	-
Enviar premio de bonificación	-	-	-	-	-	X

En cada regla se distinguen dos partes, una que corresponde a las condiciones y otra a las acciones, separadas por los cuadrantes superior e inferior. En la parte de las condiciones hay cuatro tipos de entradas: sí (S), no (N), indiferencia (-) y ningún valor.

La entrada S significa que la condición debe satisfacerse o que es cierto que la condición se satisface.

La entrada N significa que la condición no debe cumplirse o que es cierto que la condición no se cumple.

Si en una regla ninguna de las condiciones basadas en un mismo tipo de hecho de la lista de condiciones se activa (S|N) entonces estas condiciones se asumen como indiferentes (-), o sea no interesa que valor tomen.

Si en una regla alguna de las condiciones basadas en un mismo tipo de hecho de la lista de condiciones se activa (S|N) entonces el resto de las condiciones basadas en el mismo hecho no pueden tomar ningún valor.

Por ejemplo en la R3 de la Tabla 2-6 las condiciones “pago se produce en a lo sumo 30 días” y “pago se produce en al menos 21 días” son indiferentes, mientras que para la R1 la condición “pago se produce en al menos 21 días” no puede tomar ningún valor.

En una regla, en la parte de las acciones pueden existir dos tipos de entradas: la entrada ‘X’ que tiene como significado que la acción se debe realizar y la entrada ‘-’ que tiene como significado que no se realiza dicha acción.

2.2.2. Reglas dependiente e independientes

Con el objetivo de lograr un mayor entendimiento y producto de la incorporación de los cuantificadores como elementos que actúan sobre los hechos, se redefinen algunos conceptos vistos anteriormente y que son utilizados por el algoritmo que se propone más adelante.

En 2.2.1 se plantea que una condición es indiferente para una regla cuando no toma valores de si o no (S|N) y no existe una condición basada en el mismo tipo de hecho que tome algún valor de si o no (S|N) en la regla.

Reglas puras y reglas mixtas

Una regla pura es aquella que no posee condiciones indiferentes. Una regla será mixta en caso contrario. En la siguiente tabla se muestran algunos ejemplos.

Tabla2-7 Ejemplo para reglas puras y mixtas.

Condiciones	Reglas		
	R1	R2	R3
Cliente pertenece a la empresa	S	S	N
Pago se produce en al menos 31 días		S	-
Pago se produce en al menos 21 días	N		-
Cantidad solicitada supera la cantidad en stock	-	N	N
Acciones			
Acción 1	X	-	X
Acción 2	-	X	-

La regla R2 es una regla pura mientras que las reglas R1 y R3 son mixtas.

Reglas dependientes e independientes

Un par de reglas son independientes sino es posible obtener alguna regla que cumpla con las condiciones de ambas reglas, o sea si no existe un conjunto de valores que cumpla con las condiciones de ambas reglas. Un par de reglas será dependiente en caso contrario. Un par de reglas dependientes son redundantes cuando las reglas tienen las mismas entradas de acciones y contradictoria cuando tienen diferentes entradas de acciones.

Tabla2-8 Ejemplo para reglas dependientes e independientes.

Condiciones	Reglas		
	R1	R2	R3
Cliente pertenece a la empresa	S	S	S
Pago se produce en al menos 31 días	S		N
Pago se produce en al menos 21 días		S	
Cantidad solicitada supera la cantidad en stock	S	S	S
Acciones			
Acción 1	X	-	-
Acción 2	-	X	X

En la tabla anterior las reglas R2 y R3 son dependientes pues la regla “si el cliente pertenece a la empresa, el pago se produce en al menos 21 y en a lo sumo 30 días, y la cantidad solicitada supera la cantidad en stock entonces...” cumple con las condiciones tanto de R2 como de R3. Además R2 y R3 son redundantes por tener las mismas acciones. De manera similar R1 y R2 son dependientes contradictorias. Las reglas R1 y R3 son independientes.

Teniendo en cuenta estas nuevas definiciones se define el siguiente algoritmo para eliminar redundancias y detectar contradicciones.

2.2.3. Algoritmo de validación de conflictos e inconsistencias

Mediante la aplicación del algoritmo se logra hacer más compacta la tabla con la eliminación de reglas redundantes y se detectan posibles errores en las reglas por haber partido, por ejemplo, de un enunciado contradictorio.

Este proceso debe aplicarse teniendo en cuenta las condiciones que están basadas en un mismo hecho.

Como se mencionó anteriormente la tabla de decisión puede almacenar reglas de producción con un comportamiento admitido o no admitido. Las reglas que poseen un comportamiento admitido pueden incumplir con determinadas reglas de restricción, esto no las hace inconsistentes, más bien son reglas que el usuario puede definir para que se ejecuten ante comportamientos no deseados. Por lo general las reglas con comportamiento admitido ejecutan alguna acción que las diferencia del resto de las reglas (pudiera ser “lanzar error”).

El algoritmo que se propone constituye una guía en el proceso de depuración del conjunto de reglas de negocio que se analiza.

La entrada al algoritmo es un conjunto de reglas de negocio representadas como reglas de producción o reglas de restricción, como resultado se identifican contradicciones y redundancias entre ellas. Algunos problemas identificados como las contradicciones deben ser rectificadas por el usuario del negocio.

Algoritmo de validación:

Entrada: documento con el conjunto de reglas de negocio a ser analizado, específicamente reglas de producción y reglas de restricción consecuentes con los patrones definidos. Para las reglas de producción se especifica el tipo de comportamiento que poseen. De cada regla de producción se conocen las condiciones y acciones que la componen, y para cada condición se conoce que tipo de hecho la compone, si está negada y si utiliza algún cuantificador con cierta cardinalidad asociada. De cada regla de restricción se conoce que tipo de hecho restringe, si está negada y que cuantificador con su respectiva cardinalidad.

Salida: Si no existen errores no tratables se obtiene un fichero XML con las reglas depuradas. En otro caso se especifican entre que reglas se presentan los errores.

1. Captar el conjunto de reglas del fichero XML.
 - 1.1. Leer reglas de producción.
 - 1.1.1. Para cada regla de producción.

- 1.1.1.1. Identificar si debe cumplirse o no con las restricciones.
- 1.1.1.2. Identificar condiciones y acciones.
- 1.1.1.3. Para cada condición se identifican los elementos que la componen (tipo de hecho, negación, cuantificador, cardinalidad).
- 1.2. Leer reglas de restricción.
 - 1.2.1. Para cada regla de restricción
 - 1.2.1.1. Identificar los elementos que la componen (tipo de hecho, negación, cuantificador, cardinalidad).
2. Para cada regla de producción.
 - 2.1. Si posee dos condiciones basadas en un mismo tipo de hecho entonces detectar error. (Para señalar una variable con rango SBVR ofrece un cuantificador lo que a la postre se traduce en una única condición.)
3. Obtener un listado de condiciones, acciones y restricciones.
4. Crear una tabla de decisión con las dimensiones (cantidad de condiciones + cantidad de acciones, cantidad de reglas).
5. Realizar proceso de llenar la tabla de decisión.
 - 5.1. Para cada regla de producción.
 - 5.1.1. Para cada condición de la lista de condiciones.
 - 5.1.1.1. Si la condición se activa en la regla (S|N) entonces en la tabla de decisión en la posición (número de condición, regla) ubicar el valor respectivo.
 - 5.1.1.2. Si la condición no se activa en la regla y ninguna otra condición basada en el mismo hecho se activa en la regla, entonces en la tabla de decisión en la posición (número de condición, regla) ubicar el valor de indiferencia (-).
 - 5.1.2. Para cada acción de la lista de acciones.
 - 5.1.2.1. Si la acción aparece en la regla entonces en la tabla de decisión en la posición (número de condición + número de acción, regla) ubicar el valor (X).
 - 5.1.2.2. Si la acción no aparece en la regla entonces en la tabla de decisión en la posición (número de condición + número de acción, regla) ubicar el valor (-).

6. Para cada par de reglas de restricción.
 - 6.1. Si son reglas que actúan sobre un mismo hecho, equivalentes a nivel de cuantificadores (al menos 3 \equiv no - a lo sumo 2) entonces eliminar una de la lista de restricciones.
 - 6.2. Si son reglas que actúan sobre un mismo hecho y no son equivalentes a nivel de cuantificadores entonces detectar contradicción.
7. Para cada regla de restricción de la lista de restricciones.
 - 7.1. Para cada regla de producción que debe cumplir con las restricciones.
 - 7.1.1. Verificar cumplimiento de la restricción en la regla de producción.
8. Detectar contradicciones entre reglas de producción.
9. Descomponer en reglas puras las reglas de producción.
10. Eliminar reglas redundantes.
11. Solapar las reglas que cumplen con las restricciones.
12. Solapar las reglas que no tienen que cumplir con las restricciones.
13. Detectar redundancias a nivel de cuantificadores (al menos 3, a lo sumo 5).
14. Exportar reglas validadas a un archivo XML.

Los pasos del 8 al hecho 13 son explicados a continuación:

Detectar contradicciones entre reglas de producción

Se analiza todo par de reglas de producción.

Si se está en presencia de dos reglas dependientes con acciones diferentes existe una contradicción y por tanto al menos una de las reglas está mal esbozada, el experto del negocio es quien debe decidir cómo tratar esta contradicción.

Descomponer en reglas puras.

Se repite hasta que no hallan reglas mixtas.

Si una regla posee un valor de indiferencia en una condición basada en un hecho determinado entonces debe ser eliminada y se deben agregar un número de reglas igual a dos por la cantidad de condiciones de la lista de condiciones basadas en dicho hecho. A cada regla se le agrega una de las condiciones basadas en dicho hecho,

negadas o no, y hasta obtener la cantidad de reglas requeridas sin que ninguna se repita.

Luego se actualiza la lista de reglas de producción y por ende la tabla de decisión.

Eliminar reglas redundantes

Se analiza todo par de reglas de producción.

Si se está en presencia de dos reglas dependientes redundantes ambas puras donde la primera cubre al menos todos los casos de la segunda entonces la segunda regla debe ser eliminada.

Solapar reglas de producción

Se analiza todo par de reglas de producción.

Si se encuentra un par de reglas en las que se dé que tengan idénticas entradas de condiciones y acciones, excepto en una condición, donde una regla registra una entrada S y la otra registra una entrada N, ambas pueden ser eliminadas de la tabla y reemplazadas por una regla con idénticas entradas de condiciones y acciones que registren la entrada indiferente (-) en aquellas condiciones basadas en el mismo hecho donde las originales diferían.

Detectar redundancias a nivel de cuantificadores

Se analiza todo par de reglas de producción.

Si se está en presencia de dos reglas dependientes redundantes ambas puras donde ninguna cubre todos los casos de la otra entonces se detectan redundancias a nivel de cuantificadores.

Los pasos del algoritmo deben ser ejecutados en ese orden, además cada paso debe ser ejecutado mientras se encuentre algún problema. A continuación se describe con un ejemplo cómo realizar el proceso de validación.

2.2.4. Descripción del algoritmo a través de un ejemplo

Sea la siguiente tabla obtenida a partir del proceso de construcción de la tabla de decisión para un problema dado. A la izquierda se observa la lista de condiciones y la lista de acciones de la tabla de decisión. Para lograr mayor agilidad se asume que las

reglas obtenidas son de comportamiento no admitido y que cumplen con todas las reglas de restricción. Esto nos ubica en el paso 8 del algoritmo.

Tabla 2-9 Tabla de decisión obtenida a partir del proceso de construcción.

Condiciones	Reglas				
	R1	R2	R3	R4	R5
Paciente presenta VIH	S	-	S	S	N
Paciente posee filtrado glomerular ≤ 15 ml/min	N	S	S	N	S
Paciente presenta bajo índice de masa muscular	S	S	N	N	S
Paciente tiene exactamente 2 síntomas	S	N	-		
Paciente tiene al menos 4 síntomas			-	S	S
Acciones					
Acción 1	X	-	X	X	-
Acción 2	-	X	-	-	X

Antes de comenzar veamos que reglas son dependientes y cuales son independientes.

R1 y R2 independientes

R2 y R4 independientes

R1 y R3 independientes

R2 y R5 dependientes redundantes

R1 y R4 independientes

R3 y R4 independientes

R1 y R5 independientes

R3 y R5 independientes

R2 y R3 independientes

R4 y R5 independientes

Paso 8. No existen reglas dependientes contradictorias. O sea no existe un par de reglas dependientes que posean las mismas acciones, por lo que se continúa al siguiente paso del algoritmo.

Paso 9. Descomponer en reglas puras. Cada regla mixta de la Tabla 2-9 es sustituida por un conjunto de reglas puras equivalentes. A continuación se muestra la tabla de decisión anterior descompuesta en reglas puras.

Tabla 2-10 Reglas puras obtenidas.

Reglas								
R1'	R2'	R3'	R4'	R5'	R6'	R7'	R8'	R9'
S	N	S	S	S	S	S	S	N
N	S	S	S	S	S	S	N	S
S	S	S	N	N	N	N	N	S
S	N	N	S	N				
					S	N	S	S
X	-	-	X	X	X	X	X	-
-	X	X	-	-	-	-	-	X

Paso 10. Eliminar reglas redundantes. Las reglas R2' y R9' son redundantes a nivel de cuantificadores (o más bien R9' es redundante respecto a R2'), debido a esto R9' debe ser eliminada.

Paso 11. Solapar reglas con comportamiento no admitido. Las reglas R2' y R3' presentan idénticas entradas de condiciones y acciones excepto para una condición por lo tanto pueden ser solapadas. Igual sucede con las reglas {R4', R5'} y {R6', R7'}, resultado al cual se le puede aplicar nuevamente el paso. Luego de esto se obtienen las reglas mostradas en la Tabla 2-11.

Paso 12. Este paso es similar al anterior pero para las reglas no admitidas.

Tabla 2-11 Eliminación de redundancias y solapamiento de reglas.

Condiciones	Reglas			
	R1''	R2''	R3''	R4''
Paciente presenta VIH	S	-	S	S
Paciente presenta filtrado glomerular ≤ 15 ml/min	N	S	S	N
Paciente presenta bajo índice de masa muscular	S	S	N	N
Paciente tiene exactamente 2 síntomas	S	N	-	
Paciente tiene al menos 4 síntomas			-	S
Acciones				
Acción 1	X	-	-	X
Acción 2	-	X	X	-

Paso 13. No se detectan redundancias a nivel de cuantificadores y por tanto concluye el análisis de la tabla de decisión.

2.3. Conclusiones parciales del capítulo

Este capítulo se centra en la descripción de patrones de reglas de SBVR que pueden ser transformadas en reglas de producción y en el diseño de un algoritmo que se utiliza para realizar la validación de un conjunto de reglas que cumple con los patrones declarados. La exposición se divide en dos fragmentos principales. En el primer fragmento se describe el problema de la no detección de pares de reglas con anomalías que pueden ser depuradas mediante otros procesos de validación. La descripción de este problema se apoya en la utilización de algunos ejemplos de reglas. Posteriormente se describen los patrones de reglas que pueden ser capturadas para realizar la validación. El segundo fragmento se dedica a la descripción de un algoritmo para la validación de reglas que cumplen con los patrones señalados. Este abarca las siguientes etapas:

¿Cómo se realiza el proceso de construcción de las tablas de decisión a partir de un conjunto de reglas de producción?

¿Cómo se realizan los procesos de eliminación de inconsistencias y redundancias?

Luego de este desarrollo se llegan a las siguientes conclusiones:

1. La descripción de patrones de reglas modeladas en SBVR permite diseñar herramientas para su captura, esto conlleva a que se pueda realizar el análisis de la consistencia del conjunto de reglas mediante la implementación de algoritmos de validación consecuentes con los patrones definidos.
2. Se propone un algoritmo basado en tablas de decisión que permite detectar un mayor número de inconsistencias producto de la inclusión del análisis de reglas de restricción así como de la inclusión de cuantificadores para expresar cardinalidad en los hechos.

CAPÍTULO 3.

HERRAMIENTA PARA LA VALIDACIÓN DE REGLAS DE NEGOCIO

CAPÍTULO 3. HERRAMIENTA PARA LA VALIDACIÓN DE REGLAS DE NEGOCIO

Uno de los objetivos de la presente investigación consiste en validar la calidad de la herramienta sobre un conjunto de reglas. En el presente capítulo se realiza una caracterización del caso de estudio para el área de nefrología a partir de lo cual se obtiene un conjunto de reglas de negocio (ver Anexo 1) que sirven de entrada al algoritmo de análisis de consistencia propuesto. Se brinda una descripción detallada de los pasos a seguir para realizar la validación, a partir de la herramienta de software obtenida que implementa el algoritmo.

3.1. Descripción del caso de estudio

Los servicios de salud se caracterizan por un control de la evolución de los pacientes en consultas generales y, en la medida que los diagnósticos ganan en precisión y los casos así lo requieren, su continuidad en consultas especializadas. Esta práctica médica, aparentemente sencilla, supone un respaldo informativo complejo porque según cómo se comportan los casos, requieren de tratamiento diferenciado donde difícilmente se pueden abordar soluciones generales.

La Informática médica es la aplicación de la informática y las comunicaciones al área de la salud mediante el uso del software médico formando parte de las tecnologías sanitarias, tiene como objetivo prestar servicio a los profesionales de la salud para mejorar la calidad de la atención sanitaria. Es la intersección de las ciencias de la información, ciencias de la computación y la atención de la salud. Se ocupa de los recursos, los dispositivos y los métodos necesarios para optimizar la adquisición, almacenamiento, recuperación y utilización de la información en salud y biomedicina. Los instrumentos informáticos de la salud incluyen no sólo los ordenadores, sino también guías de práctica clínica, terminología médica formal, y de sistemas de información y comunicación.

Para los pacientes tratados en el área de nefrología la práctica de trasplante ha sido adoptada en los últimos años como primera opción terapéutica, en el que cada procedimiento a seguir se rige por guías que conforman complejos protocolos de

trabajo. Con los avances científicos y los resultados investigativos en este campo de la ciencia, las regulaciones para el seguimiento de casos en el área de nefrología suelen ser dinámicas, por lo que se necesitan sistemas de información capaces de adaptarse a ellas, de forma rápida y segura. En ese sentido, el enfoque de reglas de negocio resulta de gran utilidad para la obtención y administración de los requisitos y el manejo de los procesos de negocio.

El cambio constante en las reglas hace que se necesiten elementos que evalúen la calidad de las mismas de manera que se pueda detectar la presencia de contradicciones, las cuales pueden estar asociadas a elementos claves, tales como, si se debe realizar o no el trasplante renal.

Servicios del área de nefrología

En el área de nefrología la atención al paciente se realiza de acuerdo al seguimiento en tres tipos de consultas:

- Consulta de progresión o renal crónica avanzada.
- Consulta de hemodiálisis.
- Consulta de trasplante.

Cada consulta tiene requerimientos propios de información, protocolos de trabajo y restricciones, que deben ser consideradas tanto en las prácticas médicas como para los requisitos y funcionalidades que se esperan de los sistemas computacionales de apoyo. En cada tipo de consulta se registran parámetros diferentes acerca del estado de un paciente dado. Esto sugiere el manejo de un paciente como entidad abstracta, accesible desde cualquier consulta con la seguridad de que se dispone de una vista actualizada del mismo.

En la consulta de progresión o renal crónica avanzada se atiende a pacientes con enfermedad renal crónica. Está dirigida generalmente por un nefrólogo, aunque es una consulta multidisciplinaria donde participan otros especialistas como: nutriólogo, psicólogo, rehabilitador, trabajador social y cirujano. Los pacientes pueden llegar por diferentes vías (remitidos por la consulta de nefrología de su área de salud, un especialista en medicina interna o el médico de la familia, o por remisión desde el

cuerpo de guardia del hospital, la sala de nefrología u otras salas). En esta consulta son aplicados los protocolos de exámenes para determinar su estado y valorar el tratamiento. Dicho tratamiento puede recibirse por diferentes vías: en su área de salud o área de atención primaria, en la propia consulta de progresión o, si fuera inevitable, mediante métodos sustitutivos de la función renal (diálisis, hemodiálisis o trasplante renal). En caso que el paciente necesite ser tratado con alguno de los métodos sustitutivos, antes mencionados, pasa a ser atendido en las consultas propias del método, o sea hemodiálisis y trasplante.

Los posibles estadios de un paciente se encuentran exactamente entre los valores del I al V. De encontrarse el paciente en alguno de los dos primeros es reorientado a su área de salud para ser atendido allí en la consulta de nefrología. Si se encuentra en uno de los estadios III o IV el paciente es atendido en la propia consulta de progresión. Cuando el estadio es V indica que la enfermedad se encuentra en estado terminal y el paciente es remitido a uno de los métodos sustitutivos de la función renal: diálisis peritoneal, hemodiálisis o trasplante renal (TR).

Este trabajo presupone que el conjunto de reglas de negocio se ha definido en el sistema de información que da soporte a los médicos especialistas. En el Anexo 1 se especifica un conjunto de reglas obtenidas a partir del análisis del caso de estudio.

3.2. Descripción de la herramienta

En este epígrafe se abordan los principales elementos de la herramienta implementada para la validación de un conjunto de reglas de negocio que cumplen con los patrones descritos en 2.1.2. Se describe la estructura del documento que la herramienta toma como entrada, además de las principales operaciones que se pueden realizar.

3.2.1. Documento de entrada

El documento de entrada con el conjunto de reglas de restricción y reglas de producción que se desea validar, es un documento XML que debe poseer la siguiente estructura:

```
<?xml version="1.0" encoding="UTF-8"?>
<Reglas>
  <ReglaProduccionS>
    <Condiciones>
```

```
<Condicion>
  <Sujeto1/>
  <Verbo/>
  <Negacion/>
  <Cuantificador/>
  <Cardinalidad1/>
  <Cardinalidad2/>
  <Sujeto2/>
</Condicion>
...
</Condiciones>
<Acciones>
  <Accion/>
...
</Acciones>
</ReglaProduccionS>
...
<ReglaProduccionN>
  <Condiciones>
    ...
  </Condiciones>
  <Acciones>
    ...
  </Acciones>
</ReglaProduccionN>
...
<ReglaRestriccion>
  <Condiciones>
    <Condicion/>
  </Condiciones>
</ReglaRestriccion>
...
</Reglas>
```

La estructura del documento permite desglosar cada regla en los distintos elementos que la conforman.

El primer conjunto de reglas (ReglasProduccionS) corresponde a aquellas reglas de producción que deben cumplir con las reglas restricción. En el segundo conjunto de reglas (ReglasProduccionN) se especifican aquellas reglas de producción que pueden no cumplir con las restricciones. Para cada una de las reglas se presenta un conjunto de condiciones y un conjunto de acciones. El tercer conjunto de reglas (ReglaRestriccion) especifica las restricciones que limitan los aspectos del negocio.

3.2.2. Ambiente de la herramienta

La interfaz principal de la herramienta “Validación de reglas de negocio” contiene una barra de menú donde se tiene acceso a toda la funcionalidad del sistema, mediante el menú “archivo” se puede acceder a las operaciones de cargar y generar documentos de reglas, en el menú de “validación” se encuentran los principales procesos que se deben llevar a cabo para obtener un conjunto de reglas consistente, estos son, “validación de reglas de restricción”, “detección de contradicciones”, “eliminación de redundancias” y “detección de redundancias a nivel de cuantificadores”. La barra de menú también contiene un menú “ayuda” de utilidad para los usuarios sin experiencia, donde se le especifica que estructura debe tener el documento de entrada y los pasos a seguir para obtener un conjunto de reglas consistente.



Figura 3-1 Ambiente de trabajo de la herramienta Validación de reglas de negocio.

La interfaz consta de dos paneles principales para la visualización de los elementos que componen las reglas. En el panel de la izquierda se muestran las condiciones y acciones para las reglas de producción, además del conjunto de restricciones. En el panel de la derecha se muestra una tabla de decisión asociada al conjunto de reglas de producción.

3.3. Prueba de la herramienta sobre el caso de estudio

El algoritmo de validación de la consistencia propuesto en la presente investigación, permite realizar el análisis lógico del conjunto de reglas de negocio, utilizando las tablas de decisión para la representación de las mismas. A continuación y partiendo de la herramienta que implementa el algoritmo, se explica cómo obtener un conjunto de reglas consistentes. Para esto se utiliza un conjunto de reglas obtenidas a partir del caso de estudio, indicadas en el Anexo 1.

3.3.1. Preparación del documento de entrada

Para poner en práctica el algoritmo de trabajo y realizar la validación de la consistencia, se analizan grupos de reglas que tienen en común el chequeo de un conjunto pequeño de condiciones, y en dependencia de los valores de dichas condiciones realizan un conjunto finito de acciones.

Mediante una pequeña muestra de estas reglas se puede observar la estructura del documento. Se debe señalar que los cuantificadores de rango (entre ‘a’ y ‘b’) poseen dos números de cardinalidad asociada.

Reglas asociadas a trasplante

R1: Si el paciente tiene la función de los riñones menor igual que un 10%, es compatible con el donador y el donante tiene una edad mayor o igual a 18 años y menor o igual que 55, realizar trasplante renal.

Reglas de restricción

R13: En Cuba el rango de edad aceptado para donadores vivos debe ser entre 18 y 55 años.

Estructura del documento

```
<Reglas>
  <ReglaProduccionS>
    <Condiciones>
      <Condicion>
        <Sujeto1>paciente</Sujeto1>
        <Verbo>tiene</Verbo>
        <Cuantificador>a lo sumo</Cuantificador>
        <Cardinalidad1>10</Cardinalidad1>
        <Sujeto2>porcentaje de función de los riñones</Sujeto2>
      </Condicion>
```

```
<Condicion>
  <Sujeto1>paciente</Sujeto1>
  <Verbo>es compatible con</Verbo>
  <Sujeto2>donador</Sujeto2>
</Condicion>
<Condicion>
  <Sujeto1>donador</Sujeto1>
  <Verbo>posee</Verbo>
  <Cuantificador>entre</Cuantificador>
  <Cardinalidad1>18</Cardinalidad1>
  <Cardinalidad2>55</Cardinalidad2>
  <Sujeto2>años</Sujeto2>
</Condicion>
</Condiciones>
<Acciones>
  <Accion>realizar trasplante renal</Accion>
</Acciones>
</ReglaProduccionS>
...
<ReglaRestriccion>
  <Condiciones>
    <Condicion>
      <Sujeto1>donador</Sujeto1>
      <Verbo>posee</Verbo>
      <Cuantificador>entre</Cuantificador>
      <Cardinalidad1>18</Cardinalidad1>
      <Cardinalidad2>55</Cardinalidad2>
      <Sujeto2>años</Sujeto2>
    </Condicion>
  </Condiciones>
</ReglaRestriccion>
</Reglas>
```

De las reglas asociadas a trasplante (ver Anexo 1), **R5** incumple la restricción **R13** sin embargo la regla está bien diseñada y acorde con las características del proceso por lo que pasa a ser una regla con comportamiento admitido, o sea, puede incumplir con las reglas de restricción.

Una vez obtenido el documento XML con las reglas editadas se puede pasar al trabajo con la herramienta de validación y análisis de consistencia.

3.3.2. Ejecución y resultados

En esta sección se señalan los pasos a realizar en la herramienta de validación para obtener un conjunto de reglas consistente.

Una vez creado el documento con las reglas que se desean validar entonces se procede a la carga de dicho documento desde la herramienta.

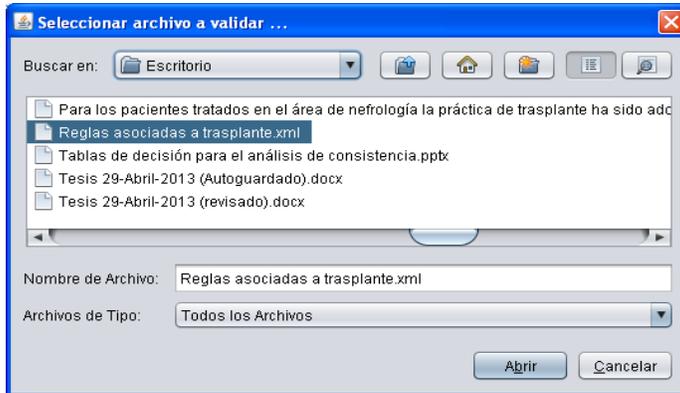


Figura 3-2 Cargar documento XML.

Ejemplo 1: Reglas asociadas a trasplante, véase Anexo 1.

Una vez cargado el documento se muestra la tabla de decisión y la lista de condiciones, acciones y restricciones asociadas.

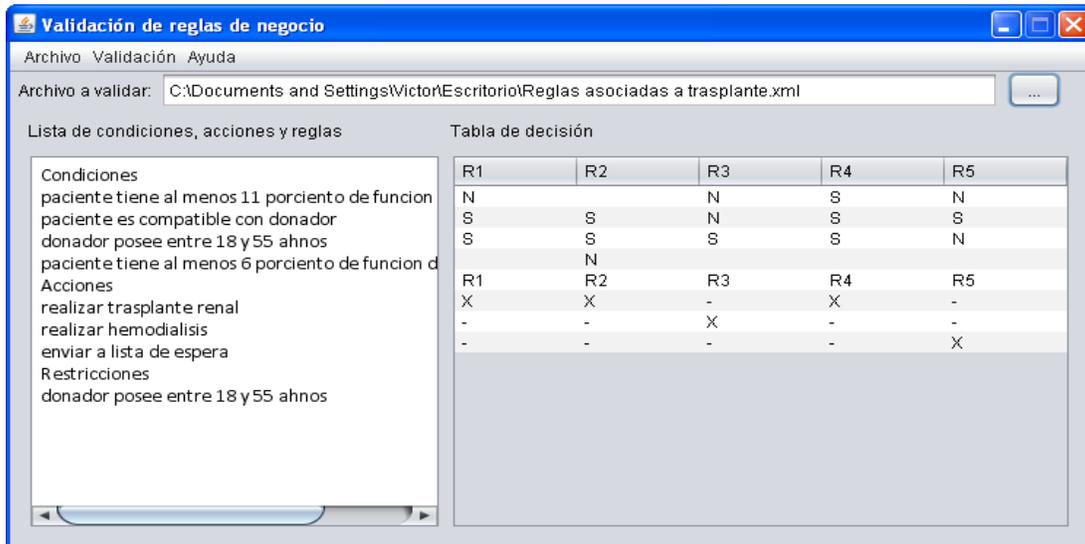


Figura 3-3 Tabla de decisión para reglas asociadas a trasplante.

Cuando las reglas han sido cargadas puede pasarse a la fase de validación. El primer análisis se realiza sobre las reglas de restricción, de estas se verifica que no presenten contradicciones entre sí y que todas las reglas de producción con comportamiento no admitido cumplan con las restricciones.

En caso de detectarse algún error entonces se muestra un mensaje indicando entre que reglas ocurre. Esto le da al usuario la posibilidad de corregirlo.

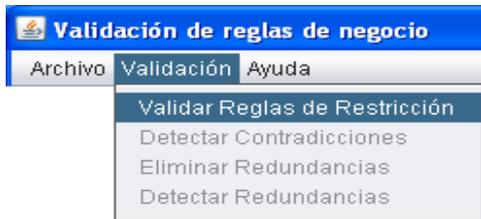


Figura 3-4 Validación de reglas de restricción.

Como no se detecta ningún error se pasa a la fase de detección de contradicciones entre reglas de producción.



Figura 3-5 Detectar contradicciones.

En caso de detectarse contradicciones se lanza un mensaje especificando entre que reglas. Como no se detectan errores se pasa a la fase de solapamiento y eliminación de redundancias.

En la Figura 3-3 se observa que la regla dos de la tabla de decisión es redundante respecto a la regla uno pues ambas reglas presentan idénticas entradas de condiciones y acciones excepto para un par de condiciones basadas en el hecho “paciente posee porcentaje de funcionalidad de los riñones” donde se tiene que para:

R1: paciente no posee al menos 11 por ciento de funcionalidad de los riñones

R2: paciente no posee al menos 5 por ciento de funcionalidad de los riñones

La condición que presenta R2 está implícita en la condición que presenta R1, haciendo a R2 redundante respecto a R1 y por tanto posible de eliminar.

Por otro lado R1 y R4 presentan idénticas entradas de condiciones y acciones excepto para una condición donde una posee una entrada ‘S’ y la otra una entrada ‘N’, por tanto pueden ser solapadas. La siguiente figura muestra la tabla de decisión que puede ser obtenida luego de eliminar las redundancias y realizar el solapamiento.

Tabla de decisión

R1	R2	R3	R4	R5
N		N	S	N
S	S	N	S	S
S	S	S	S	N
	N			
R1	R2	R3	R4	R5
X	X	-	X	-
-	-	X	-	-
-	-	-	-	X

=

Tabla de decisión

R1	R2	R3
-	N	N
S	N	S
S	S	N
-		
R1	R2	R3
X	-	-
-	X	-
-	-	X

Figura 3-6 Eliminación de redundancias y solapamiento de reglas.

Luego se pasa a la detección de redundancias a nivel de cuantificadores que no pueden ser eliminadas pues requieren la definición de nuevas condiciones. Si existe alguna redundancia de este tipo se le especifica al usuario entre que par de reglas ocurre.

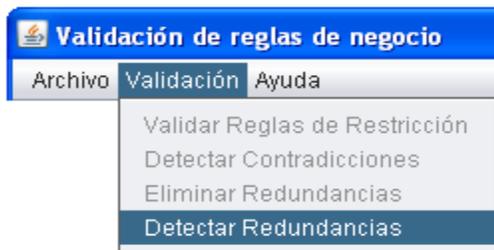


Figura 3-7 Detectar redundancias entre cuantificadores.

Si todos los pasos anteriores son ejecutados correctamente entonces se puede obtener un nuevo conjunto de reglas consistente, lo que se realiza mediante la opción exportar reglas que genera un documento XML con las reglas validadas.

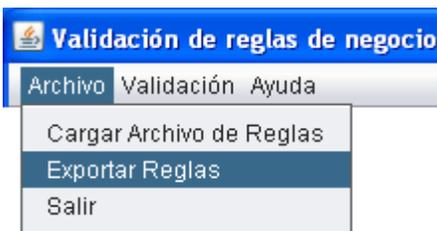


Figura 3-8 Exportar reglas a documento XML.

Ejemplo 2: Reglas asociadas a remitidos a consultas, véase Anexo 1.

A continuación se inicia el proceso de validación para el conjunto de reglas definido.

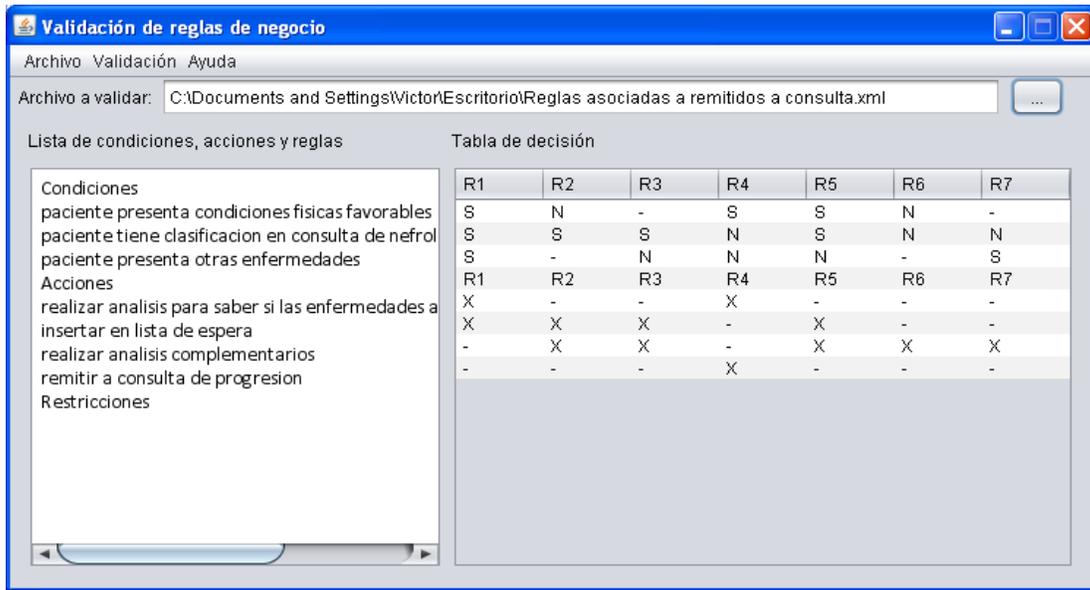


Figura 3-9 Tabla de decisión para reglas asociadas a remitidos a consulta.

Una vez cargado el archivo de reglas se pasa a la fase de validación. Al no detectarse restricciones (no hay reglas de restricción) se ejecuta el siguiente paso.

Las reglas no presentan contradicciones y por tanto se pasa a la eliminación de redundancias y el solapamiento de las reglas. De este proceso se obtienen las siguientes reglas:

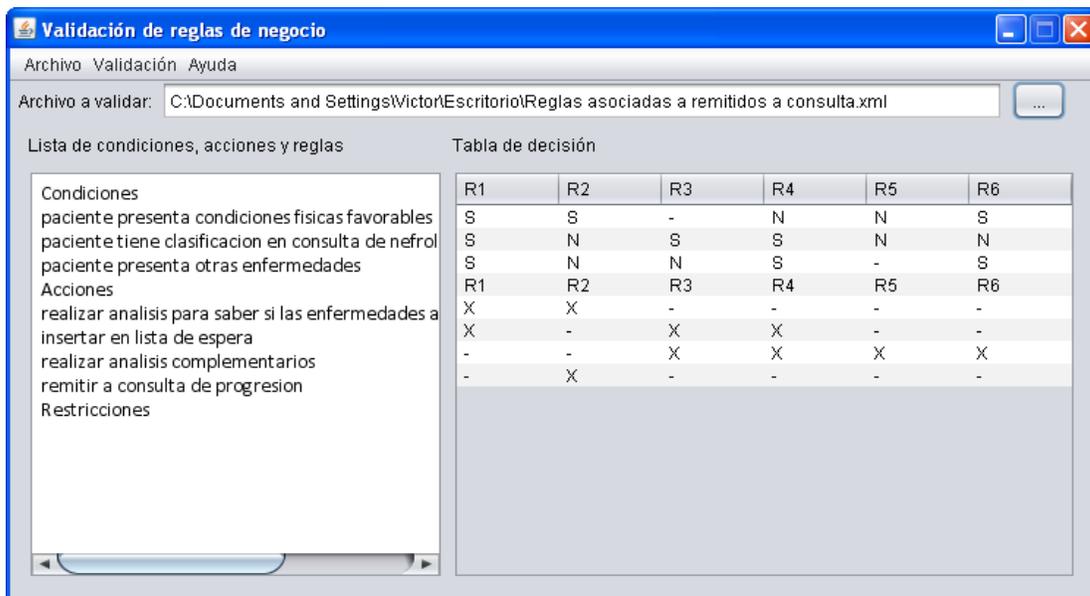


Figura 3-10 Eliminación de redundancias y solapamiento de reglas.

Como no se detectan otras anomalías puede entonces generarse un archivo con las reglas resultantes. Para este caso se obtiene un conjunto de reglas consistente.

3.4. Conclusiones parciales del capítulo

En este capítulo se realiza la aplicación del algoritmo de validación de reglas de negocio basado en tablas de decisión al caso de estudio del área de nefrología.

1. Se aplica el algoritmo para el caso de estudio, ilustrando mediante ejemplos como lo realiza la herramienta implementada en esta investigación.
2. Mediante el algoritmo implementado se detectan contradicciones a partir del análisis de las restricciones, y anomalías del tipo ambivalencia, redundancias por columnas duplicadas, columnas subsumidas, fila de condición redundante y redundancias entre cuantificadores, producto del análisis de las reglas de producción.

Conclusiones

Como resultado de esta investigación se tiene que la definición de patrones de reglas expresadas en SBVR, y sus transformaciones a otros mecanismos de validación permite obtener un conjunto de reglas consistente. Particularmente se tienen las siguientes conclusiones:

1. Se identifican las anomalías que se pueden detectar mediante las tablas de decisión.
2. Se describen patrones de reglas expresadas en SBVR que pueden ser representadas mediante tablas de decisión.
3. Se proponen transformaciones para expresar el conjunto de reglas de negocio desde SBVR a tablas de decisión.
4. Se propone un algoritmo basado en tablas de decisión que realiza el análisis de consistencia y chequeos de integridad de un conjunto de reglas que cumple con los patrones definidos y que detecta anomalías del tipo ambivalencia, redundancias por columnas duplicadas, columnas subsumidas, fila de condición redundante y entre cuantificadores.
5. Se obtiene una herramienta que realiza el análisis de consistencia y chequeos de integridad mediante la implementación del algoritmo propuesto.
6. Se valida la calidad de la herramienta sobre un conjunto de reglas para el caso de estudio Trasplante Renal área de nefrología.

Recomendaciones

1. Describir otros patrones de reglas que puedan ser analizados mediante el algoritmo de validación.
2. Ampliar las capacidades del algoritmo para la detección de otras anomalías que pueden ser introducidas mediante la incorporación de otros conceptos de SBVR como los roles que poseen los términos o las características asociadas a los tipos de hecho.
3. Dar continuidad al estudio mediante la incorporación de otros métodos para el análisis de la consistencia del conjunto de reglas de negocio.

Referencias Bibliográficas

- Amaolo, M. P. (2011) Algunas consideraciones para la transformación de Semántica de Negocios SBVR en el Lenguaje de Ontologías Web OWL2.
- Appleton, D. S. (1984) Business Rules - The Missing Link. *Datamation*, 30, 145-150pp.
- Bailey, J., Bry, F. & Pátránjan, P.-L. (2005): Composite event queries for reactivity on the web. Special interest tracks and posters of the 14th international conference on World Wide Web. *Última visita: 1082-1083p*
- Bajec, M. (2001): Definition of the conceptual framework for BRM in organisations. Laboratory of Information Systems, UL Faculty of Computer & Information Science. Eslovenia. *p*
- Bajec, M. & Krisper, M. (2005) A methodology and tool support for managing business rules in organisations. *Information Systems*, 30, 423-443pp.
- Bajec, M., Krisper, M. & Rupnik, R. (2000): Using Business Rules Technologies To Bridge The Gap Between Business and Business Applications. The IFIP 16th World Computer Congress 2000, Information Technology for Business Management. Peking, China. RECHNU, G. E. 77-85pp
- Bajec, M. & Marjan, K. (2001) Managing business rules in enterprises. *Electrotechnical Review*, 68, 236-241pp.
- Barnes, M. & Kelly, D. (1997) Play by the rules. *BYTE*, 22, 98-102pp.
- Barwise, J. & Etchemendy, J. (1999): *Language, Proof and Logic*.pp
- BRG (2000): *Defining Business Rules ~ What Are They Really?* Business Rules Group. Seattle. 77pp
- BRG (2003): *The Business Rule Manifesto - The Principles of Rule Independence*.pp
- Castelo, C. A. M. (2011): *Investigación para la adopción y uso de los motores de reglas de negocio*. San Buena Ventura. Santiago de Cali. *p*
- Castilla, M. J. (2007) *Sistemas de Información II*.
- Ceri, S. & Fraternali, P. (1997): *Designing Database Applications with Objects and Rules: The IDEA Methodology*. Addison-Wesley. pp
- Chappel, O. (2005) Term-Fact Modeling, the Key to Successful Rule-Based Systems. *Business Rules Journal*, 6.
- Date, C. J. (2000): *What Not How: The Business Rules Approach to Application Development*. Addison-Wesley Longman Publishing Co. Boston, MA. pp
- Goedertier, S. & Vanthienen, J. (2005): Rule-based Business Process Modeling and Execution. IEEE EDOC Workshop on Vocabularies Ontologies and Rules for The Enterprise (VORTE 2005). Septiembre.

- Goedertier, S. & Vanthienen, J. (2006): Rule-based business process modeling and execution. Department of Decision Sciences & Information Management Naamsestraat 69 - 3000 Leuven. Katholieke Universiteit Leuven. *p*
- Hay, D. C., Kolber, A. & Anderson, K. H. (1997): GUIDE Business Rules Project: Final Report.
- Landinez, I. (2009): Español Estructurado. *Última visita:*
- Martínez Busto, M. E., Moreno Montes de Oca, I., Bishoksan, K., Caballero Martínez, A., Boggiano Castillo, M. B., Pérez Alonso, A. & González González, L. (2010): Authoring, managing and implementing business rules related to kidney transplantation using droolstab plug-in for protégé. Memorias del Workshop on Machine Learning and Knowledge Discovery, Cuba-Flanders. Santa Clara, Villa Clara, Cuba. 2-6 de Febrero.
- Martínez Fernández, J. L. (2010): Introduciendo Semántica en un proceso de Desarrollo Software a través de Reglas de Negocio. Universidad Politécnica de Madrid. *p*
- Martínez, R. E. B., Ramírez, N. C., Mesa, H. G. A., Suárez, I. R., Trejo, M. d. C. G., León, P. P. & Morales, S. L. B. (2009) Árboles de decisión como herramienta en el diagnóstico médico.
- MEJIA, C. A. (2011): Investigación para la adopción y uso de los motores de reglas de negocio. Universidad San Buenaventura. Santiago de Cali. *p*
- Moreno Montes de Oca, I., González González, L. & Martínez Busto, M. E. (2008): Modelo de hechos genérico. CENDA UCLV. Cuba.
- Morgan, T. (2002): Business Rules and Information Systems: Aligning IT with Business Goals. Addison Wesley. *pp*
- Núñez, L. (2012): Validación de la consistencia de un conjunto de reglas de negocio. Ciencia de la Computación. Universidad Central de Las Villas. Santa Clara. *p*
- OMG (2003): Business Semantics of Business Rules - Request For Proposal. *Última visita:* Julio de 2009
- OMG (2008): Semantics of Business Vocabulary and Business Rules (SBVR), v1.0.*pp*
- Rosca, D., Greenspan, S. J., Wild, J. C., Reubeinstein, H. B., Maly, K. & Feblowitz, M. (1995): Application of a Decision Support Mechanism to the Business Rules Lifecycle. KBSE. 114-121*pp*
- Ross, R. G. (2000): What are Fact Models and why do you need them? . <http://www.brcommunity.com/cgi-bin/x.pl/commentary/b008a.html>, <http://www.brcommunity.com/cgi-bin/x.pl/commentary/b008b.html>. *Última visita:*
- Ross, R. G. (2003): Principles of the Business Rule Approach. Addison-Wesley Professional. *pp*

- Ross, R. G. (2005a) Decision Tables, Part 1 ~ The Route to Consolidated Business Logic. Business Rules Journal, 6.
- Ross, R. G. (2005b) Decision Tables, Part 2 ~ The Route to Completeness. Business Rules Journal, 6.
- Ross, R. G. (2009): Business Rule Concepts. 33-37 pp
- Ross, R. G. (2010): Business Rule Concepts. Business Rule Solutions, LLC. 158 pp
- Santos, Y. (2012): Validación de consistencia basada en la lógica formal de reglas de negocio Ciencia de la Computación. Universidad Central de Las Villas. Santa Clara. p
- Soria, E., Serrano, A. J. & Martín, J. D. (2009): Métodos de ayuda al diagnóstico clínico. Tema 6: Árboles de decisión.
- Struck, D. L. & Willshire, M. J. (1999): Business rule continuous requirements environment. Colorado Technical University. Colorado Springs, Colorado. p
- Van Der Aalst, W. M. P. (1998) The application of Petri nets to workflow management. The journal of Circuits, 8, 21-66pp.
- Vanthienen, J. (2008): SBVR: The ABCs of Accurate Business Communication. <http://www.BRCommunity.com/a2008/b403.html>. Última visita:
- Vanthienen, J. (2012): The History of Modeling Decisions using Tables <http://www.BRCommunity.com/a2012/b637.html>
<http://www.BRCommunity.com/a2012/b641.html>
<http://www.BRCommunity.com/a2012/b652.html>. Última visita:
- Weiden, M., Hermans, L., Schreiber, G. & van der Zee, S. (2004): Classification and Representation of Business Rules. www.omg.org/docs/ad/02-12-18.pdf. Última visita:
- Witt, G. (2009a) A Practical Method of Developing Natural Language Rule Statements (Part 1). Business Rules Journal, 10.
- Witt, G. (2009b) A Practical Method of Developing Natural Language Rule Statements (Part 2). Business Rules Journal, 10.
- Witt, G. (2009c) A Practical Method of Developing Natural Language Rule Statements (Part 3). Business Rules Journal, 10.
- Youdeowei, A. (1997): The B-Rule Methodology: A Business Rule Approach to Information Systems Development. Department of Computation UMIST. Manchester, United Kingdom. p
- Zaidi, A. K. & Levis, A. H. (1995) Validation and Verification of Decision Making Rules. Symposium on Analysis, Design and Evaluation of Man-Machine Systems.

Anexo 1: Reglas obtenidas para el área de nefrología

Reglas asociadas a trasplante

R1: Si el paciente tiene la función de los riñones menor igual que un 10%, es compatible con el donador y el donante tiene una edad mayor o igual a 18 años y menor o igual que 55, realizar trasplante renal.

R2: Si el paciente tiene la función de los riñones menor igual que un 5%, es compatible con el donador y el donante tiene una edad mayor o igual a 18 años y menor o igual que 55, realizar trasplante renal.

R3: Si el paciente tiene la función de los riñones menor igual que un 10%, no es compatible con el donador, y el donante tiene una edad mayor o igual a 18 años y menor o igual que 55, realizar hemodiálisis.

R4: Realizar trasplante renal si el paciente tiene la función de los riñones mayor que un 10%, es compatible con el donador y el donante tiene una edad mayor o igual a 18 años y menor o igual que 55.

R5: Si el paciente tiene la función de los riñones menor igual que un 10%, es compatible con el donador y el donante no tiene una edad mayor a 18 años y menor o igual que 55, enviara lista de espera.

Reglas asociadas a remitidos a consultas

R6: Si el paciente presenta condiciones físicas favorables, se conoce su clasificación en consulta de nefrología y presenta otras enfermedades, se le realiza análisis para saber si dichas enfermedades afectan el TR y se inserta en la lista de espera.

R7: Si el paciente presenta condiciones físicas no favorables y se conoce su clasificación en consulta de nefrología se le realiza análisis complementarios y se inserta en la lista de espera.

R8: Si el paciente no padece de otras enfermedades y se conoce su clasificación en consulta de nefrología se le realiza análisis complementarios y se inserta en la lista de espera.

R9: Si el paciente presenta condiciones físicas favorables, no padece de otras enfermedades y no se conoce su clasificación en consulta de nefrología, se le realiza análisis de enfermedades que afectan el TR y se remite a consulta de progresión.

R10: Si el paciente presenta condiciones físicas favorables y no padece de otras enfermedades y se conoce su clasificación en consulta de nefrología se le realiza análisis complementarios y se inserta en la lista de espera.

R11: Si el paciente no presenta condiciones físicas favorables y no se conoce su clasificación se le realizan análisis complementarios.

R12: Si el paciente padece de otras enfermedades y no se conoce su clasificación se le realizan análisis complementarios.

Reglas de restricción

R13: En Cuba el rango de edad aceptado para donadores vivos debe ser entre 18 y 55 años.