

Universidad Central "Marta Abreu"
de Las Villas
Facultad Matemática, Física y Computación
Departamento de Ciencia de la Computación



*TESIS EN OPCIÓN DEL TÍTULO ACADÉMICO DE
MASTER EN CIENCIAS TÉCNICAS*

Validación de la consistencia de un conjunto de reglas de negocio

Autor: Lic. Lester Núñez Pérez

Tutores: M.Sc. María Elena Martínez Busto

Santa Clara

Junio 2012

"Año 54 de la Revolución"

Ser cultos es el único modo de ser libre

José Martí

Esta tesis la dedico a toda mi familia, en especial a mi madre, mi abuela, mi hermanita, a mi padre, a todas mis tías y mi tío, y especialmente a mi esposa Anne que con su esfuerzo y dedicación han hecho posible mi realización profesional y superación.

*A todos los que han estado durante todo este tiempo a mi lado,
que han confiado en mí.*

*A mis profesores que más que tutores los considero mis amigos, tesoros que he ganado en
toda mi trayectoria y de los cuales me siento orgulloso...*

*En especial a mi tutora María Elena que siempre me ha apoyado en el plano profesional y
a la cual le debo este trabajo.*

A todos sepan que los llevo sinceramente en un lugarcito especial de mi corazón.

Resumen

En la presente investigación se realiza un estudio sobre formas de validar la consistencia de un conjunto de reglas de negocio (RN) y la existencia de una metodología. Es la Metodología PROTEUS quien propone pasos para todo el ciclo de vida de las RN. La validación se ubica en las etapas 6 y 11 de esta. Se analizan, además, diferentes métodos de validación de la consistencia de RN, son ellos: español estructurado, árboles de decisión 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. Este algoritmo se aplica al conjunto de reglas existentes del caso de estudio área de nefrología, y se comprueba que detecta determinados tipos de anomalías.

El producto final es una herramienta independiente de la base de datos que realiza la validación de la consistencia de un conjunto de reglas de forma automática, basado en el algoritmo confeccionado sobre tablas de decisión.

Abstract

In the present investigation is a study on ways to validate the consistency of a set of business rules and the existence of a methodology. PROTEUS methodology is the one who proposes steps for the entire lifecycle of business rules. Validation is at 6 and 11 stages of this methodology. They also analyze different methods of validating the consistency of business rules, are they: Spanish structured decision trees and decision tables. As a result of the analysis method is chosen based on decision tables considering the advantages it offers. Drawing up an algorithm that validates the consistency of a set of rules based on decision tables. This algorithm is applied to all existing rules of the case study area of nephrology, and checked to detect certain types of anomalies.

The final product is a separate tool from the database to perform validation of the consistency of a set of rules automatically, based on the algorithm made on decision tables.

Tabla de contenido

Introducción.....	9
Capítulo I: Validación del repositorio de reglas de negocio.....	6
1.1 Enfoque de reglas de negocio.....	6
1.1.1 Definición y propiedades	7
1.1.2 Ciclo de vida	9
1.1.3 Modelo de hechos y ontologías.....	14
1.1.4 Valoración de clasificaciones de reglas de negocio	16
1.2 Metodología PROTEUS.....	19
1.3 Estándar para la representación del vocabulario y reglas de negocio (SBVR).....	22
1.4 Validación de la consistencia de reglas de negocio	25
1.4.1 Árboles de decisión.....	26
1.4.1.1 Características y ventajas de los árboles de decisión.....	26
1.4.1.2 Identificación de los requerimientos.....	28
1.4.2 Español estructurado.....	28
1.4.2.1 Declaraciones estructuradas.....	29
1.4.3 Tablas de decisión.....	30
1.4.3.1 Definición de Tabla de Decisión como estructura algebraica.....	30
1.4.3.2 Estructura de una Tabla de Decisiones	31
1.4.3.3 Tipos de reglas y equivalencias.....	32
1.4.3.4 Tipos de anomalías identificables.....	34
1.4.3.5 Depuración de las tablas de decisión.....	35
1.5 Conclusiones parciales del capítulo	37
Capítulo II: Métodos de validación de la consistencia sobre tablas de decisión	38
2.1 Patrones de reglas de negocio.....	38
2.2 Tabla de decisión para la validación del conjunto de reglas.....	43
2.2.1 Construcción de las tablas de decisión	44
2.2.2 Construcción de la tabla de decisión	46
2.2.3 Eliminación de inconsistencias y ambigüedades en el conjunto de reglas.....	48

2.2.4	Otros tipos de tablas de decisión.....	53
2.3	Herramienta PROLOGA.....	55
2.3.1	Anomalías identificadas.....	56
2.3.2	Anomalías del tipo inter-tabular.....	57
2.3.3	Anomalías del tipo intra-tabular.....	58
2.3.4	Otros tipos de anomalías.....	62
2.4	Análisis de ambas vías para validación de la consistencia usando tablas de decisión.....	64
2.5	Algoritmo para la validación de la consistencia del conjunto de reglas de negocios.....	68
2.5.1	Estructura del documento de entrada.....	68
2.5.2	Descripción general.....	69
2.5.3	Descripción detallada.....	69
2.6	Tipos de reglas que se puede aplicar el algoritmo.....	71
2.7	Conclusiones parciales del capítulo.....	72
Capítulo III: Validación de reglas de negocio para el caso de estudio.....		74
3.1	Caso de estudio: área de nefrología.....	74
3.1.1	Descripción.....	75
3.1.2	Caracterización del caso de estudio.....	76
3.2	Captura del conjunto de reglas usando UCRules.....	77
3.2.1	Herramienta UCRules.....	78
3.2.2	Conjunto de reglas de negocio.....	79
3.3	Aplicación del algoritmo de validación para el caso de estudio.....	80
3.3.1	Conjunto de reglas de negocio.....	80
3.3.2	Construcción de la tabla de decisión.....	81
3.3.3	Depuración de la tabla de decisión.....	81
3.4	Corrida del algoritmo implementado.....	87
3.5	Conclusiones parciales del capítulo.....	95
Conclusiones.....		96
Recomendaciones.....		97
Bibliografía.....		98
Anexos.....		104
Anexo 1: Clasificación de reglas de negocio según diferentes criterios.....		104
Anexo 2: Pseudocódigo del algoritmo de validación de la consistencia de un conjunto de reglas.....		107

Introducción

La identificación de todas las regulaciones relacionadas con un servicio dado y su posterior consideración en sistemas que las hagan cumplir, constituye sin lugar a dudas un nuevo enfoque que eleva el rigor y mejora la calidad de los servicios que se ofrecen.

Las aplicaciones computacionales que le dan soporte requieren renovaciones y adaptaciones para cumplir con las necesidades siempre cambiantes de estos tiempos. Las reglas de negocio (RN) pueden ser usadas como un instrumento para desarrollar aplicaciones flexibles y bases de datos modificables con facilidad (Bajec, 2001, Bajec and Marjan, 2006, Bajec et al., 2000, Barne and Kelly, 1997, Date, 2000a, Youdeowei, 1997).

Las RN son definiciones explícitas que regulan cómo opera un determinado negocio y cómo el mismo es estructurado (Bajec, 2001, Bajec and Marjan, 2006, Bajec et al., 2000, Barne and Kelly, 1997, Date, 2000a, Youdeowei, 1997); se entiende por negocio cualquier tipo de servicio que ofrezca determinada institución. Estas RN son imprescindibles para el funcionamiento de la empresa o institución correspondiente, así como para el sistema de información que soporta sus procesos.

Existen diferentes tipos de RN, entre las formas más simples se encuentran las tradicionales validaciones que restringen valores válidos para los datos que se captan, tal es el caso por ejemplo que el sexo de un paciente puede ser sólo femenino o masculino; no obstante no es este el tipo de regla que es de interés en el enfoque de RN, sino aquellas que constituyen normas, regulaciones, lineamientos, etc., que son representativas del negocio y sensibles a ser modificadas (Bajec and Krisper, 2005).

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 RN, los desarrolladores y los usuarios tienen la posibilidad de definir las reglas centralmente previo al desarrollo de las aplicaciones, logrando de esta forma una independencia de las aplicaciones que las usan, reutilizar dichas reglas, modificarlas y administrarlas, facilitándose así su

mantenimiento y 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 and Krisper, 2005).

El enfoque de RN aborda esta problemática; en él 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.

El problema de captar y modelar las RN es inherente tanto a investigadores académicos como al personal de la industria (Bailey, 2005). Deben desarrollarse herramientas variadas para lograr su mantenimiento; sin embargo, actualmente, en el Desarrollo de los Sistemas de Información (DSI), las RN casi no se emplean en la forma en que son definidas en las investigaciones de este tema. O sea, no son separadas de los componentes irrelevantes para el conocimiento del negocio.

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. Pero esto no es suficiente ya que una vez capturadas pueden no reflejar ideas precisas o incluso pueden resultar contradictorias entre sí, por tanto se requiere hacer un tratamiento adicional al conjunto de reglas creado, realizar una validación de las mismas.

Es común que en un enfoque de RN, la estructura de un sistema se visualice, por medio del Modelo de Hechos que modela las relaciones entre conceptos, y se enfoca en la estandarización de la terminología del negocio para establecer un vocabulario de negocio, lo que es considerado un punto de partida crucial en la modelación basada en este enfoque (Ross, 1997a, Nilsson, 1999, Appelton, 1984a, Bajec, 2006, Bajec, 2000, Barne, 1997, BRG, 2000a, BRG, 2003b, Ceri and Fraternali, 1997a, Date, 2000b, Hendryx, 2003, Hüsemann and Schäfer, 2006, Morgan, 2002, Ross, 2003, Ross, 2005, Ross and Lam, 2003, Struck, 1999a, Gottesdiener, 1999, Youdeowei, 1997, Rosca et al., 1997b, Herbst et al., 1994a, Hay and Healy, 1997) (Bailey, 2005).

En el establecimiento de dicho vocabulario para representar y comunicar conocimiento de un tema y el conjunto de interacciones que existen entre los términos de ese vocabulario han tenido éxito las ontologías, a partir del cual se conforma “un glosario de

términos ontológicos” (Bailey, 2005) al proporcionar una comprensión compartida del conocimiento de un dominio que puede ser comunicada entre personas y sistemas heterogéneos (Bailey, 2005).

Se necesita entonces que el conjunto de reglas ofrezca ciertas facilidades tales como:

- Mostrar las reglas en una interfaz amena y sencilla.
- Permitir adicionar comentarios referentes a cada regla.
- Posibilitar la edición y modificación de reglas.
- Realizar validación y chequeo de integridad al añadir reglas al conjunto.
- Confirmar la aceptación de las reglas añadidas manteniendo la consistencia del sistema.

Para que el conjunto de reglas ofrezca las facilidades anteriormente mencionadas se realiza manejo de ontologías (Bailey, 2005). Se emplea una ontología para representar el Modelo de Hechos usando esta herramienta lo cual posibilita exportar las reglas en formato XML, apropiado para posteriormente realizar las validaciones y chequeos de integridad necesarios, garantizando así la consistencia del conjunto lo cual justifica el planteamiento del problema de investigación siguiente:

Existen procesadores de RN que permiten a los usuarios, o lo hacen de una forma muy rudimentaria, plantear las regulaciones del negocio en un lenguaje cercano al natural, que luego se almacenan en un conjunto al cual, generalmente, no se le realiza validaciones de la consistencia ni chequeos de integridad tan necesarios para mantener la integridad del sistema que garantice su procesamiento posterior.

El **objetivo general** de la investigación consiste en proponer una vía que permita realizar validaciones de la consistencia y chequeos de integridad sobre un conjunto de reglas de negocio, a través del análisis lógico de las reglas para eliminar diferentes tipos de anomalías.

Este se desglosa en los siguientes **objetivos específicos**:

1. Realizar un estudio de métodos de validación de la consistencia para reglas de negocio.

2. Valorar diferentes métodos para la validación de la consistencia y el chequeo de integridad sobre el conjunto de reglas de negocio.
3. Proponer una solución para realizar la validación de la consistencia y el chequeo de integridad del conjunto representado sobre el modelo de hechos genérico con enfoque ontológico.
4. Aplicar los resultados de la investigación al caso de estudio.
5. Confeccionar una herramienta que detecte determinadas anomalías en el conjunto de reglas.

Las preguntas de investigación planteadas son:

¿Qué métodos existen para realizar la validación de la consistencia y el chequeo de integridad de un conjunto de RN?

¿Cómo se realiza la validación de la consistencia de RN por métodos existentes?

¿Cuál es el método que se puede aplicar al caso de estudio?

¿Cómo diseñar el algoritmo, basado en el modelo de representación de reglas de negocio ya existente, que permita la validación de la consistencia y el chequeo de integridad del conjunto de RN?

¿Cómo confeccionar una herramienta que detecte determinadas anomalías en el conjunto de reglas?

Después de haber realizado el marco teórico se formuló la siguiente *hipótesis de investigación* como presunta respuesta a las preguntas de investigación:

H1. Un algoritmo desglosado en un conjunto de pasos basado en un modelo de representación de reglas de negocio ya existente permite realizar validaciones de la consistencia y chequeos de integridad del conjunto de reglas de negocio para eliminar determinadas anomalías en el mismo.

Para lograr los objetivos trazados y demostrar la hipótesis de investigación se acometieron las **tareas de investigación** siguientes:

- Investigar sobre métodos existentes de validación de la consistencia de RN.

- Ubicar dentro de la metodología PROTEUS[®] los pasos donde se realiza la validación de las RN.
- Proponer una vía para realizar validación de la consistencia y chequeo de integridad del conjunto de RN.
- Confeccionar una herramienta que realice la validación de la consistencia y chequeo de integridad del conjunto de RN según la vía propuesta.
- Ejemplificar los resultados de la investigación para el caso de estudio médico.

El valor práctico del trabajo está dado por la necesidad de disponer de un algoritmo para realizar la validación de la consistencia y el chequeo de integridad de un conjunto de RN.

La tesis está estructurada en tres capítulos. En el Capítulo 1 se aborda el enfoque de RN, diferentes métodos de validación de la consistencia de RN existentes y en qué parte de la metodología PROTEUS[®] se realiza dicha validación. En el Capítulo 2 se trata sobre el método de validación de la consistencia de RN basado en tablas de decisión, enfatizando en dos vías ya existentes de realizarla. En el Capítulo 3 se aplican los resultados teóricos de la investigación en el caso de estudio: *área de nefrología* y la implementación de una herramienta que detecta determinados tipos de anomalías en el conjunto de reglas. Este documento culmina con las conclusiones, recomendaciones, referencias bibliográficas y los anexos.

Capítulo I: Validación del repositorio de reglas de negocio

Actualmente se desarrollan muchos proyectos de investigación sobre el tema de las RN, su análisis, clasificación, articulación, formulación y documentación (Goedertier and Vanthienen, 2006b, Bailey et al., 2005, Microsoft, 2003, Keller et al., 1992, Wilkes and Veryard, 2004, Halpin, 2000b, Antoniou et al., 2004). Una de las áreas más prominentes es la integración del ambiente de negocio de la Organización y su Sistema de Información (SI) de apoyo (Rosca, 1997, Bajec and Marjan, 2006, Dobson, 1992.). En este capítulo aborda el enfoque de RN, se hace un breve recorrido hasta llegar al concepto, características fundamentales, clasificación y ciclo de vida de las reglas. Se aborda la metodología Proteus desarrollada por un grupo de investigadores entre los que se encuentra Ronald G. Ross (Ross, 2008). Finalmente se aborda un marco de trabajo propuesto por (Zachman, 1987, Zachman, 2002) y se establecen las etapas dentro del ciclo de vida de las RN, enfatizando en la validación de RN basada en tablas de decisión.

1.1 Enfoque de reglas de negocio

La primera aparición de las RN en el contexto de los sistemas de información fue en 1984 cuando Daniel Appelton escribió un artículo llamado “Reglas de Negocio: el enlace perdido” (Appelton, 1984b), argumentando que ellas residen en la línea divisoria entre la ingeniería del negocio y la ingeniería de software, es decir, entre el negocio y el sistema que lo soporta (Weiden et al., 2002). Appelton discutió problemas que eran causados por la falta de estandarización de términos de negocios. Su punto de vista consistía en que los analistas del negocio no podían proporcionar soluciones comunes si los usuarios usaban términos que variaban en significado de un departamento a otro dentro de una misma organización.

Luego de la aparición de este trabajo (Appelton, 1984b), muchos profesionales de la computación y de los SI comienzan a manejar la idea de que si las RN podían ser capturadas en máquinas de procesamiento (como lo son las reglas en los sistemas expertos), entonces dichas máquinas podrían hacer cumplir estas reglas y asegurar que los procesos de negocio fuesen controlados y conducidos de acuerdo a los estándares del mismo, sus políticas y procedimientos (Struck, 1999a). Este fue solo el comienzo. Las RN siempre han sido controladas por el negocio, pero en las últimas décadas la responsabilidad de administración y control de su cumplimiento ha

sido delegado generalmente a los desarrolladores de software, los que las han incorporado dentro de los diseños de los sistemas de cómputo.

El enfoque de RN está dirigido a proporcionar a las personas del negocio un control directo sobre el funcionamiento del mismo. Ofrece muchas facilidades a la hora de desarrollar aplicaciones, las hace más flexibles y de gran adaptabilidad. Tanto investigadores como ejecutores, están convencidos de que las RN demandan un tratamiento especial durante el Desarrollo de Sistemas de Información (DSI) (Youdeowei, 1997, Date, 2000a, Bajec and Krisper, 2005, Ross, 1997b, Petrounias and Loucopoulos, 1994), para lo cual es necesario hablar el mismo idioma entre los expertos del negocio y los desarrolladores. Las reglas deben ser captadas de forma automática en los SI, y separadas de la lógica de las aplicaciones para facilitar su ulterior mantenimiento (Bajec, 2001, Bajec and Marjan, 2006, Bajec et al., 2000, Barne and Kelly, 1997, Date, 2000a, Youdeowei, 1997). 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, 1999c, Struck, 1999b, Goedertier and Vanthienen, 2006b).

1.1.1 Definición y propiedades

Según Ross¹ (Ross, 1997b) “los propietarios del negocio tratan las RN de una manera diferente a los desarrolladores”. Para los primeros no son más que directivas que tienden a influenciar o guiar el comportamiento del negocio, mientras que para los últimos estas son vistas como estructuras dispersas en la lógica del programa, que no siempre surgen desde las políticas del negocio. Desde el surgimiento del término RN los autores abordan su definición de forma diferente

(Appelton, 1984b, Bajec et al., 2000, BRG, 2000b, Ceri and Fraternali, 1997b, Hay, 1997, Morgan, 2002, Rosca et al., 1995, Ross, 2003, Youdeowei, 1997, Ross, 2006, Castilla, 2008), algunas carecen de claridad, por lo que solamente se van a citar las de mayor precisión.

¹ Es reconocido internacionalmente como “el padre de las reglas de negocio”. Ha organizado la Conferencia Forum de Reglas de Negocio desde 1997, fundador del Grupo de Reglas de Negocio (BRG) en los años 1980, autor-editor de importantes artículos del BRG, tales como: “*The Business Motivation Model: Business Governance in a Volatile World*” (2007) y “*Business Rules Manifesto*” (2003). Es también miembro activo de la OMG, donde dirige el desarrollo de estándares para RN y modelos del negocio, incluido SBVR.

Morgan², un distinguido profesor, plantea lo siguiente acerca de las RN:

“Una RN puede considerarse una declaración compacta sobre un aspecto del negocio, usando un lenguaje simple, inequívoco, accesible a todas las partes interesadas: el propietario del negocio, el analista, el arquitecto técnico, desarrolladores, entre otros” (Morgan, 2002).

El propio autor expone una mejor caracterización de las RN al considerarlas compactas que abarcan un aspecto del negocio. Morgan introduce por primera vez la idea de expresarla utilizando un lenguaje, que además debe ser simple, inequívoco y accesible por todos. Además define cuáles son los autores involucrados con ellas.

Por su parte Ross en (Ross, 2003, Ross, 2006) logra ser muy preciso, basado fundamentalmente en lo planteado por Morgan (Morgan, 2002) y Bajec (Bajec et al., 2000), al plantear su definición de RN:

“Una RN 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 RN más detalladas.*
- *En caso de ser reducida perdería información importante sobre el negocio.”*

Ross refleja en su definición que las reglas deben ser atómicas: *no pueden ser descompuestas sin que pierdan información*, término también novedoso.

Después de analizar cada uno de estos criterios podemos proponer que las RN son sentencias que definen o restringen aspectos del negocio, estableciendo la estructura del mismo. Son accesibles a todas las partes interesadas del negocio usando un lenguaje simple, inequívoco y común, y cumplen con las siguientes propiedades (Marisel Mass Chaviano and García, 2010):

² Profesor distinguido en Ciencia de la Computación en la Universidad de Neumont, donde ha impartido docencia en niveles de pregrado y posgrado, además de investigar en el futuro de los sistemas de información. Tiene una vasta experiencia práctica en relación a la creación y desarrollo de SI. Ha trabajado en compañías tales como EDS y Unisys. Su interés técnico principal está en el uso de manejadores de modelos para construir sistemas de información que evidencien el valor de negocios reales. Autor de numerosos artículos y libros que abarcan la temática de las RN.

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

Se observa que las RN son estructuras individuales que permiten ser definidas, delimitadas y expresadas de forma precisa, modificable y que en su conjunto componen el marco estructural, la política, la estrategia y la operatividad de una organización. Son accesibles a los usuarios expertos empleando un lenguaje simple e inequívoco. Estas características responden a las siguientes preguntas sobre RN a las que se refiere Ross en (Ross, 2003):

Qué: ¿Qué es la regla? ¿Qué las caracteriza? Se plantea que son *elementos individuales, definidos, delimitados*.

Cómo: ¿Cómo pueden ser modeladas? Se plantea que son *expresadas de forma inteligible y componen el marco estructural, la política, la estrategia y la operatividad*.

Dónde: ¿Dónde son válidas? ¿Cuál es su jurisdicción? Se propone que son válidas para el marco de una *organización*.

Quién: ¿Quién es el responsable de ellas? ¿Por quién y cómo son accesibles? Se expresa que son accedidas por *usuarios expertos empleando un lenguaje simple e inequívoco*.

Por qué: ¿Cuál es la razón de ser de esta regla? Mediante ella se define una organización desde su estructura, políticas, estrategias y operaciones.

La idea principal del *enfoque de RN* es que las reglas deben estar separadas de los datos y funcionalidades de las aplicaciones. No obstante ellas deben ser de fácil acceso, visibilidad, modificación y administración, tanto para desarrolladores como para los propietarios del negocio (Bajec et al., 2000).

1.1.2 Ciclo de vida

Se definen para el ciclo de vida de las RN varias actividades, entre las que se encuentran las siguientes (Goedertier and Vanthienen, 2006a, López G., 2006): Descubrimiento, Análisis, Validación de conflictos y consistencia, Modelación, Implementación, Identificación de las reglas y Mantenimiento.

La figura 1.1 muestra diferentes etapas del ciclo de vida de las reglas relacionadas con el desarrollo de sistemas de información (DSI) guiado por reglas.



Figura 1.1: Desarrollo de Sistemas de Información (DSI) basados en RN

Existen tareas que permiten el mantenimiento de las RN en una organización y que son independientes del DSI, estas son: la identificación de las reglas de acuerdo a la clasificación y mantenimiento propiamente dicho. A continuación se amplían cada una de las actividades antes mencionadas.

Descubrimiento

El propósito del descubrimiento es analizar toda la información sobre la estructura del SI que se diseña, lo que constituye una fuente de información para la derivación de los llamados negocios oscuros (*business rumbling*) (Goedertier and Vanthienen, 2006a, Herbst, 1995, Ross, 2008). Este término, negocios oscuros, es una pieza no estructurada de información, relacionada con aspectos específicos de un negocio pero que carece de una estructura, resultando como consecuencia un término informal, ambiguo e inconsistente.

La documentación para la especificación de requisitos que incluye información acerca de actividades que ocurren en el negocio, la naturaleza de los datos usados, la cultura de trabajo y la ética de la organización, los miembros claves de la organización, etc., resultan ser la entrada al descubrimiento de las RN.

Los Negocios pueden también ser deducidos a partir de las personas del negocio, a través de técnicas tradicionales de especificación de requerimientos, tales como entrevistas, cuestionarios, observación del trabajo y monitoreo ad-hoc de los métodos de trabajo. Sin

embargo, si la organización mantiene el modelo de negocio el conjunto inicial de reglas puede ser derivado directamente a partir de la información captada dentro del modelo, lo que sería más fácil y de mayor rigor.

Análisis

Los Negocios Oscuros son descritos en lenguaje natural y pueden contener más de una regla de negocio simple. A través del análisis de las RN, son analizados y descompuestos en sentencias de información discreta, atómica y precisa, también llamadas RN (Ross, 2008).

El proceso de descomposición incluye su clasificación, que puede basarse en unos de los esquemas de clasificación. El propósito de un esquema de clasificación es proporcionar estructuras dentro de las cuales las RN puedan ser fijadas respecto a la naturaleza de la información que transmiten. Actualmente existen varias taxonomías capaces de clasificar las RN (Bailey et al., 2005, Goedertier and Vanthienen, 2006a, Halpin, 2000a, Hamadi and Benatallah, 2003, Keller et al., 1992, Martínez Busto et al., 2006, Meredith and Bjorg, 2003, Van Der Aalst et al., 2003)

Validación de conflictos y consistencia

La validación de conflictos y consistencia es una de las tareas más delicadas y consumidoras de tiempo en las investigaciones de RN, tanto como la semántica de las reglas, especificada en el lenguaje natural, que resultan esenciales para comprender y desarrollar los SI (Vanthienen, 2008, Vasilecas and Lebedys, 2007). En las RN cada palabra tiene que convenir en significado preciso, no puede tener varios significados entre los especialistas del negocio.

En el descubrimiento de las RN la organización es dividida en unidades manejables. No obstante, diferentes áreas de la organización tienen que responder al mismo tipo de eventos de negocio y tienen información similar, necesaria para completar estos eventos; estas áreas usualmente no comparten el mismo vocabulario, procedimientos y métodos para localizar los recursos que les permiten cumplir su misión (Ross and Lam, 2006, Vanthienen, 2008). Es por esta razón que las reglas, desde diferentes unidades, muchas veces pueden ocasionar conflictos con otras y de esta manera ser inconsistentes.

La validación de conflictos y consistencia se comienza en el modelo de negocios, y se repite cada vez que se adiciona una regla que se descubre en el desarrollo de un SI particular. Esta es la etapa, dentro del ciclo de vida de las RN, que es de interés de esta investigación.

Modelación

La modelación de las RN es con frecuencia un proceso iterativo y heurístico. Las reglas al principio son solo oraciones que expresan políticas generales formales y específicas, que comúnmente describen de manera general e imprecisa, son solo los expertos los encargados de traducirlas en sentencias específicas y significativas. Sin embargo, estas sentencias aún se consideran divagaciones del negocio, sin disciplina, algunas veces son claras y otras ambiguas, la mayoría de las veces contienen más de una idea (Von Halle, 1994, Halpin, 2007). Todo esto es un calificador de la importancia que tiene la etapa de modelado de las RN.

En contraste con las investigaciones tradicionales, donde el modelo conceptual es desarrollado enteramente desde la documentación de especificación de requisitos, ahora se avoca el uso de la documentación de RN como la posibilidad de acceder a lo fundamental y se propone establecer un enlace entre las RN y los objetos de información asociados. El enlace resulta ser esencial para comprender el *por qué* un objeto de información específico es definido y usado de una manera concreta.

Implementación

Las reglas que gobiernan un negocio por si solas no se pueden controlar fácilmente sin embargo su automatización las hace muy potentes. Es precisamente la implementación una de las actividades más importantes en el DSI basado en RN. Actualmente existen potentes tecnologías que pretenden soportar el mantenimiento y la implementación de las RN en un SI (incluida la generación de código) (Akal et al., 2005, Barne and Kelly, 1997).

Existe un, ya amplio, rango de herramientas orientadas a base de datos (BD) que se esfuerza en usar mecanismos de BD usando reglas, tales como disparadores y procedimientos almacenados, para que sistemas orientados a reglas ofrezcan, además de un lenguaje de especificación de reglas declarativas y procedimientos, mecanismos especiales para llevar a cabo la ejecución de las mismas.

Uno de los retos de implementación de las RN es proporcionar herramientas que soporten la generación de código de programa (para la generación automática de RN) a partir de las especificaciones de las reglas. Las herramientas que actualmente son capaces de ofrecer este tipo de funcionalidad son llamadas Orientadas a Tecnología (Boggiano Castillo et al., 2009b, BRG, 2003a). Ellas ofrecen generación de código pero solamente para tecnologías seleccionadas. Por ejemplo si se usa una herramienta que trabaja con una máquina de reglas el código de programa puede solamente ser generado para ser usado con esta máquina de regla. Sin embargo, a partir de las especificaciones, que deben ser rigurosamente interpretadas, pueden ser computadas; de esta forma queda limitado el número de categorías de reglas que puede ser procesado.

Una promisorio idea para la solución de este problema lo es la Arquitectura Dirigida por Modelo (Model Driver Architecture, MDA) (Goedertier and Vanthienen, 2006a). El MDA define dos tipos de modelos: independientes de la plataforma (PIM) y los de plataforma específica (PSM).

Identificación

La identificación de las reglas operacionales, tácticas y estratégicas es típicamente hecho en relación con el modelado de las RN. Mientras algunas de las reglas requieren determinaciones explícitas (ejemplo las reglas que son asociadas con los objetivos del negocio y procesos de negocio) muchos otros son inherentes en los diagramas de modelo de negocio. Tales reglas pueden ser automáticamente extraídas desde el modelo de negocios y entrar en su potencial representación en la determinación de requerimientos de un SI particular.

Mantenimiento

Las RN son muy dinámicas, están en constante cambio debido a que los negocios son muy cambiantes en respuesta a fuerzas externas tales como regulaciones, acciones por competidores, nuevos descubrimientos científicos, etc, o por fuerzas internas cuando un objetivo previamente establecido cambia. El propósito del manteniendo de las RN es coordinar estos cambios de forma tal que el sistema pueda reconsiderar sus objetivos de forma consistente y ajustables acordes a las estrategias del negocio. Es importante que las RN sean mantenidas sobre el nivel de la organización y no sobre un SI particular. Las RN

pueden tener una implementación o varias (en uno o varios SI) pero desde el punto de vista de la organización ellas siempre reflejan el mismo ambiente de negocio. El mantenimiento de las RN incluye algunas actividades fundamentalmente: control de la eficiencia, control de versiones, validación de conflictos y consistencia y el control de impacto.

Las RN deben definirse y mantenerse de forma independiente a los modelos y los procesos con los que la empresa funciona. Aunque entre ellos se mantenga una relación íntima y constante.

1.1.3 Modelo de hechos y ontologías

La administración de las RN se hace posible cuando se crean estructuras para modelar las mismas. Dichas estructuras precisan y facilitan la manipulación de las reglas. El trabajo del analista consiste en manipular un conjunto de formas estructurales, predefinidas como base para generar la representación textual equivalente, 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 tiene una definición clara y precisa desde la perspectiva del negocio y debe ser reflejada en el glosario de términos, que es la colección de todos los términos y sus definiciones, lo que hace que el modelo de hechos sea mucho más general que un glosario. Esta es la base de un principio importante del *enfoque de RN* “se trabaja mejor si todos hablan el mismo lenguaje” (Ross, 2003).

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: es indivisible.
- Cognoscible: representa cosas que existen, conocimiento acerca del negocio.

Cada conexión lógica entre conceptos tiene una forma estándar, estas conexiones son conocidas como *hechos*. 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 1.2).

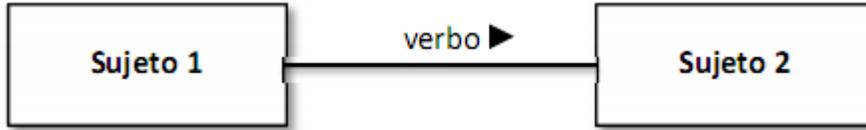


Figura 1.2: Representación gráfica abstracta de un hecho (Moreno Montes de Oca, 2008, Martínez Busto et al., 2010b)

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 1.3 se muestra la relación que existe entre los términos, hechos y reglas en el modelo.

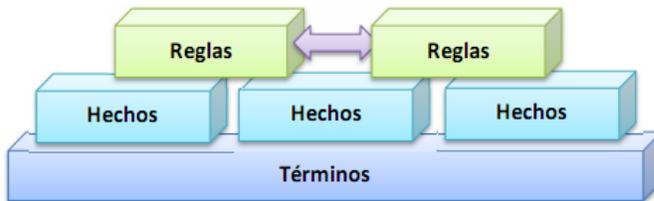


Figura 1.3: Pirámide de Términos, Hechos y Reglas (Chappel, 2005, Moreno Montes de Oca et al., 2008, Martínez Busto et al., 2010a)

El modelo 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). Cuando uno o más conceptos están relacionados se definen como *Tipos Fácticos* (Morgan, 2002, Spreeuwenberg and Gerrits, 2006) y forman la base para las expresiones de reglas. Esto se hace de manera similar a como se definen los conceptos en los modelos de ontologías.

Las ontologías son modelos que representan una abstracción del dominio de manera formal, de modo tal que varias partes son capaces de coincidir en la abstracción y reúso del modelo para sus propias aplicaciones (Amaolo, 2007). Por tanto una ontología constituye la estructura que organiza el conocimiento en forma sistemática. En este caso se utiliza para la captura de la semántica de las RN (Widom and Ceri, 1996, Von Halle, 2002).

Una ontología se define como un acuerdo parcial y explícito de una conceptualización. Mientras que Gruber (1995) la define como una especificación explícita de una conceptualización. Una ontología está conformada por conceptos (identificados como relevantes en el dominio a modelar), sus atributos, relaciones entre conceptos, funciones y axiomas. Los axiomas son especificados para obtener una mejor caracterización de la semántica.

Desde el punto de vista de los SI una ontología es una herramienta de representación de conocimiento que facilita la expresión de conceptos y las relaciones que existen entre estos conceptos, incluyendo contenido semántico no ambiguo y prestando especial atención a la necesidad de compartir esta representación semántica entre distintos agentes (Martínez Fernández, 2010). Tanto el vocabulario del negocio como una ontología se fundamentan en conceptos y reglas interrelacionados (ejemplo: identidad, cardinalidad o taxonomía) que restringen y especifican la comprensión del significado de los conceptos (Spreeuwenberg and Gerrits, 2006).

Normalmente una taxonomía de términos, sin ninguna definición no se considera una ontología. Sin embargo, en el caso de la ontología de empresa se hace de esta forma debido a que se anticipa que las definiciones serán incorporadas luego, y los términos están diseñados para corresponder a ellas en la pre-implementación (Uschold and Gruninger., 1996). Este concepto se corresponde con las necesidades del modelo de hechos para el *enfoque de RN*.

1.1.4 Valoración de clasificaciones de reglas de negocio

La clasificación de las reglas facilita el descubrimiento, análisis y finalmente la modelación de las mismas. Clasificar las reglas garantiza la consistencia de la descripción y eleva su claridad. El propósito de dicha tarea, como etapa dentro del análisis, es facilitar la identificación y lograr su atomicidad.

Generalmente las RN son descritas mediante un lenguaje de reglas semi-formal o formal, considerando que un lenguaje natural no es idóneo para este nivel. Resulta apropiado el empleo de diferentes lenguajes de modelación, como pueden ser: español o inglés estructurado, UML, OCL (acrónimo del inglés: Object Constraint Language) o Método de Ross (Ross, 1997b), código de programa, entre otros (Núñez Pérez et al., 2010). De todo lo anterior se infiere que definir plantillas o patrones para cada categoría de RN facilita el trabajo.

En esta dirección varios autores proponen taxonomías de clasificación según el nivel de abstracción, funcionalidad, relación con el SI, entre otros criterios (Ross and Lam, 2006, Kovacis et al., 2002, Morgan, 2002, Weiden et al., 2002) (Véase Anexo 1). Este epígrafe aborda una de estas taxonomías por considerarla relevante para la presente investigación.

Clasificación Semántica

La clasificación según la semántica de la regla es un esquema propuesto por Weiden en (Weiden et al., 2002) que es usado para clasificar las RN de acuerdo a sus propiedades semánticas, definiendo tipos de RN, agrupados en tres categorías (Weiden, 2000, Weiden et al., 2004): *estructurales*, *de comportamiento o conducta* y *de administración*. Estas categorías representan diferentes visiones del negocio. Las reglas estructurales son las concernientes a la descripción de aspectos estáticos de un negocio. Las de comportamiento definen las condiciones sobre la ejecución de tareas en el negocio. La categoría de reglas de administración define las restricciones de alto nivel sobre el negocio.

Los puntos de vista: *estructural* y *de comportamiento*, definen la visión interna del proceso de negocio. El punto de vista *administrativo* introduce nociones tales como objetivo, valor, recursos necesarios de tareas, actores y procesos del negocio (Weiden, 2000). Este esquema de clasificación asume que se ha creado el modelo de procesos inicial del negocio en términos de los procesos y tareas principales.

Cada una de las tres categorías es subdividida en varios tipos de RN:

Estructural: estructura de conceptos, persistencia e historia.

De Conducta: flujo de información, pre-condición, post-condición, frecuencia, duración, flujo de control y conocimiento de la tarea.

Administrativa: organización, objetivo y valor, actitud del actor, responsabilidad del actor y recursos.

Este esquema se apoya en la semántica de las reglas, proporciona categorías, que guardan un estrecho vínculo con los elementos del modelo del negocio, por lo que se considera cercano a los propietarios. A diferencia de la taxonomía propuesta por Goedertier³ en (Goedertier et al., 2007a,

³ Es Dr. en Ciencias Económicas Aplicadas en 2008, investigador en administración de procesos de negocio. Profesor de numerosos cursos doctorales, tales como: introducción a la metodología de la investigación, lenguajes

Martínez Fernández, 2010), que también se apoya en la semántica de las reglas desde la óptica de las personas del negocio y proporciona muchas sub-categorías, alejándose del negocio y dando una visión más cercana al programador.

Las razones antes expuestas hacen que en esta investigación se utilice como guía la clasificación semántica propuesta por Weiden (Weiden et al., 2004). Esta se complementa con la clasificación cercana a la implementación que se aborda a continuación.

Clasificación cercana a la implementación

Reportada por Soliveres en (Soliveres, 1997), donde se acerca a la visión del implementador y propone las categorías de RN siguientes:

Reglas del modelo de datos: engloba todas aquellas reglas que se encargan de validar la información básica almacenada para cada atributo o propiedad de una entidad u objeto.

Reglas de relación: incluye todas aquellas reglas que controlan las relaciones entre los datos.

Reglas de restricción: restringe los datos que el sistema puede contener. Nótese que este grupo de reglas se solapa, en cierto modo, con las *reglas del modelo de datos*, dado que aquellas también impiden la introducción de datos erróneos. La diferencia estriba en que este tipo de regla restringe el valor de los atributos o propiedades de una entidad más allá de las restricciones básicas que sobre las mismas existen.

Reglas de derivación: aquellas reglas que permiten derivar cierta información a partir de otra, controlan la obtención de información, realización de cálculos, etc.

Reglas de flujo: son las que determinan y limitan cómo fluye la información a través de un sistema, indican qué camino recorre la información y obligan a que se sigan solo aquellos que son permitidos.

Esta taxonomía se vincula estrechamente con la forma en que las RN son implementadas en el SI, dando la visión de los desarrolladores. Las tres primeras categorías conforman un grupo que

y metodologías de la investigación, aprendizaje automático e inferencia inductiva, entre otros. Revisor de numerosas publicaciones, entre las que se encuentra: Information Systems sobre modelación de procesos de negocio. Es miembro del comité del programa del evento "The International RuleML Symposium 2008", sobre el intercambio y aplicación de reglas.

está dirigido a la estructura y validación de los datos. Considerando este grupo como una categoría, y unida a los dos restantes se observa gran similitud con las clasificaciones propuestas por Kovacic (Kovacic et al., 2002) y Goedertier (Goedertier et al., 2007a, Goedertier and Vanthienen, 2005), diferenciándose básicamente en que estas dan una visión más cercana al negocio.

1.2 Metodología PROTEUS

Una de las tareas más costosas en el proceso de implantación de un SI basado en RN es la identificación de estas reglas a partir de todas las fuentes de información del negocio que se manejan, regulaciones, documentos, manuales, etc. Disponer de una metodología que simplifique este proceso de extracción de las reglas resultaría de gran utilidad. No se encontró ninguna metodología proporcionada por algún organismo de estandarización, aunque PROTEUS[®] (Ross, 2006, Ross and Lam, 2006) es una de las más conocidas y es independiente de las herramientas seleccionadas para la gestión de las RN.

PROTEUS[®] (Ross, 2006, Ross and Lam, 2006) es una metodología que proporciona una serie de pasos y técnicas que facilitan la captura, expresión y organización de RN. Sus principales características son:

- Organización, expresión y captura exhaustiva de RN.
- Orientación al negocio.
- Sirve de guía para orientar el proceso de definición de requisitos.
- Permite construir un modelo de negocio con la participación directa del usuario.
- Contempla el descubrimiento de RN a partir de los entregables del modelo de negocio.
- Actúa de manual en el que los desarrolladores de software pueden consultar dudas sobre el negocio.

El modelo de negocio mencionado se compone a su vez de:

- Política o estrategia de trabajo, un plan de acción para la solución de negocio buscada.
- Modelos de flujo de trabajo, planificando respuestas a eventos de negocio relevantes, definiendo tareas y responsabilidades.

- Modelo de Hechos, indicando el conocimiento compartido sobre las operaciones de negocio que resulta necesario para la operativa del mismo.
- Un catálogo de conceptos donde se define el vocabulario estándar correspondiente al área de negocio en cuestión.
- Y, por supuesto, las RN.

Aunque la metodología está orientada al negocio, es necesario proporcionar un conjunto de requisitos completo y exhaustivo que permita a los arquitectos de sistemas diseñar el sistema necesario. Se consideran las siguientes acepciones para los conceptos básicos a manejar en la metodología (Martínez Fernández, 2010):

- **Negocio:** una organización económica, cualquier actividad o tipo de actividad sistemática y con sentido.
- **Regla de negocio:** regla que está bajo la jurisdicción del negocio.
- **Concepto:** Algo concebido en la mente como pensamiento, idea, noción que incluye todo aquello que característicamente se asocia o sugiere un término.
- **Catálogo de conceptos:** Listado de los contenidos de un vocabulario de negocio estructurado.
- **Modelo de hechos:** Una visualización o guía para un vocabulario de negocio estructurado.
- **Tipo de hecho:** Característica específica que puede conocerse sobre una o más cosas y que es importante para la operativa del negocio.
- **Regla:** Guía para una conducta o acción. Elemento de un conjunto de regulaciones, habitualmente oficiales, a través del cual se rige una actividad.
- **Término:** Una palabra o expresión que tiene un significado perfectamente delimitado.

Aparte de lo mencionado, cuando se define una regla deben tenerse en cuenta otros aspectos como el ámbito en que la regla tiene validez, los actores que intervienen en la misma, el proceso de mantenimiento que es donde entra a jugar su papel la validación que es el aspecto fundamental de esta investigación; las causas de la definición de la regla, etc.

La metodología PROTEUS[®] distingue dos grandes etapas en el proceso de desarrollo de RN. En la primera de ellas se define un modelo de negocio que contempla tanto la especificación del dominio en que se desenvuelve el negocio como la detección de las RN que deben reflejarse para obtener el producto o servicio deseado.

La figura 1.4 muestra el flujo de actividades que contempla la metodología para la especificación del modelo de negocio. En ella se especifica que en las etapas 6y 11 se realizan análisis de las reglas donde el autor considera que es donde se debe realizar la validación de las reglas para determinar si existen inconsistencias. Se pueden describir algunos aspectos que resultan comunes a cualquier proceso de identificación y definición de RN. En el paso 4 se habla de términos y hechos, es donde se define el vocabulario sobre el que se van a definir las reglas. Puede parecer que las fases 6 y 8 son redundantes pero no es así si se tiene en cuenta que la fase 8 pretende recoger reglas específicas para un producto o servicio, mientras que las fases de la 4 a la 6 se dedican a establecer el modelo de negocio genérico sobre el que se van a desarrollar todos los productos o servicios.

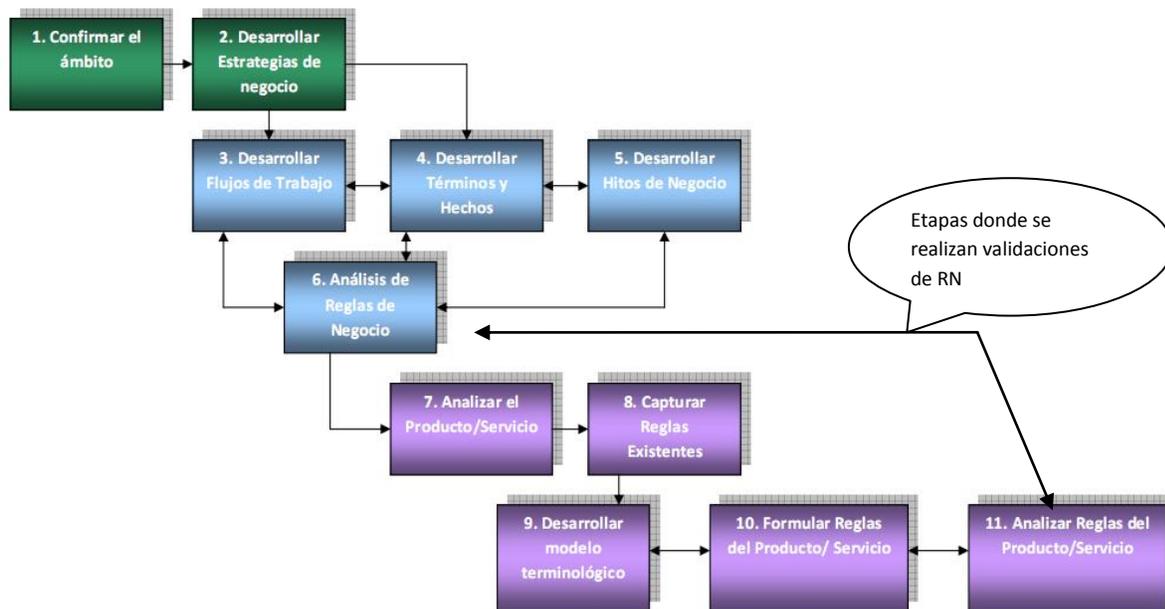


Figura 1.4: Fases de metodología PROTEUS[®] para modelado de negocio (Martínez Fernández, 2010)

A modo de información, en la metodología se definen una serie de etapas para construir el que denomina modelo de sistema, es decir, la implementación del modelo de negocio definido en la

primera etapa. Para representar las RN en el modelo de negocio el BRG⁴ crea un lenguaje denominado *Semantic Business Vocabulary and Rules (SBVR)* (BRG, 2009) que se describirá en el siguiente epígrafe.

1.3 Estándar para la representación del vocabulario y reglas de negocio (SBVR)

El lenguaje SBVR ha sido definido por el grupo dedicado al estudio de las RN llamado BRG como respuesta a la solicitud de propuestas (*Request For Proposal RFP*) (Martínez Fernández, 2010, BRG, 2009) realizada por el *Object Management Group (OMG)* y denominada *Business Semantics for Business Rules (BSBR)*.

El SBVR es 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 incluyen también 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 MDA⁵ como se muestra en figura 1.5.

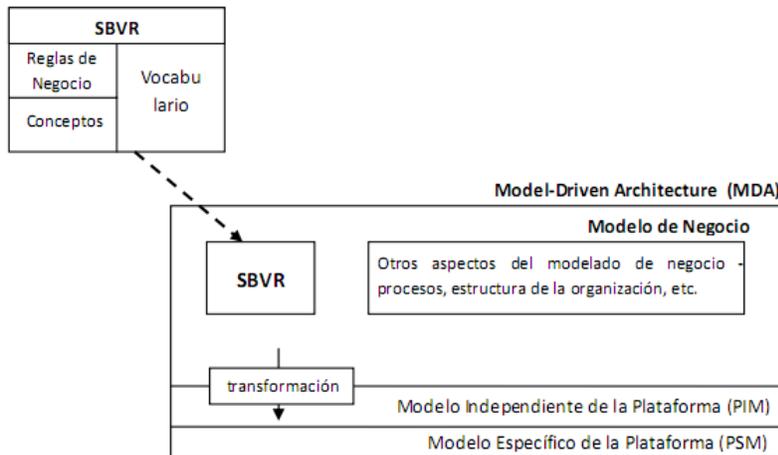


Figura 1.5: SBVR en el marco de la arquitectura MDA (OMG, 2008)

Esta ubicación de SBVR en la arquitectura MDA tiene dos implicaciones:

⁴ *Business Rules Group (BRG)* es un grupo de investigadores creado en 1989, interesado en resaltar el papel de las reglas de negocio en la construcción de un Sistema de Información.

⁵ Arquitectura Dirigida por Modelos (MDA) se basa en la especificación de modelos para los componentes de software que sean capaces de procesar datos automáticamente, dichos modelos recogen las especificaciones o requisitos del sistema que se quiere construir.

- Se centra en reglas y vocabularios de negocio, especialmente en aquellos relevantes para ser utilizados junto con las reglas. Es también necesario desarrollar otros aspectos del modelo de negocio, incluyendo procesos de negocio y la estructura de la organización.
- Los modelos de negocio describen el negocio y no los sistemas de información que los soportan.

La semántica en el SBVR se define como “el significado o relación entre significados de un signo o un conjunto de signos”. Estos signos pueden tener cualquier forma: palabras, frases, códigos, números, iconos, sonidos, etc. SBVR incluye dos vocabularios especializados:

- El “Vocabulario para Describir Vocabularios de Negocio”, que contempla todo tipo de términos y significados, aparte de los de las RN.
- El “Vocabulario para describir RN”, que se centra en la especificación del significado de las RN y se construye sobre el vocabulario anterior.

Ambos vocabularios han sido separados, de manera que el “Vocabulario para Describir Vocabularios de Negocio” pueda ser utilizado de forma independiente, por ejemplo, como base para vocabularios de procesos de negocios o roles organizativos (BRG, 2009).

El vocabulario de negocio contiene todos los elementos especializados y definiciones de conceptos que una organización o comunidad utilizan. Un vocabulario basado en SBVR potencia la semántica de los glosarios de términos de negocio habituales y sus definiciones.

Uno de los aspectos esenciales a tener en cuenta en el tratamiento que SBVR da al concepto de *regla* es su consistencia con la lógica formal. Así, en SBVR, una regla es *un elemento de orientación que introduce una obligación o una necesidad*, ejemplos de ello son las reglas estructurales que expresan necesidades y las reglas operativas que expresan obligaciones (Martínez Fernández, 2010).

El metamodelo SBVR está diseñado para proporcionar interfaces e intercambios de datos estandarizados entre herramientas que recogen, organizan, analizan y utilizan reglas y vocabularios de negocio. Podrá ser soportado por múltiples herramientas que proporcionan validación, análisis, alineación, combinación y composición de RN, y para el intercambio de vocabularios y RN junto con su semántica. Una característica importante del metamodelo es que comienza con los vocabularios acompañado de un conjunto de reglas que permite transformar el

vocabulario a MOF (*Meta-data Object Family*) / XMI (*Meta-data Interchange*), lenguajes estándar para la representación de modelos.

La formulación lógica del SBVR proporciona una sintaxis formal, abstracta e independiente del lenguaje para la asociación de un conjunto de significados compartidos a los vocabularios empleados por las comunidades⁶. Estable una correspondencia con el estándar XMI, que permite el intercambio de conceptos, hechos y RN entre herramientas que soportan SBVR (ver figura 1.6).

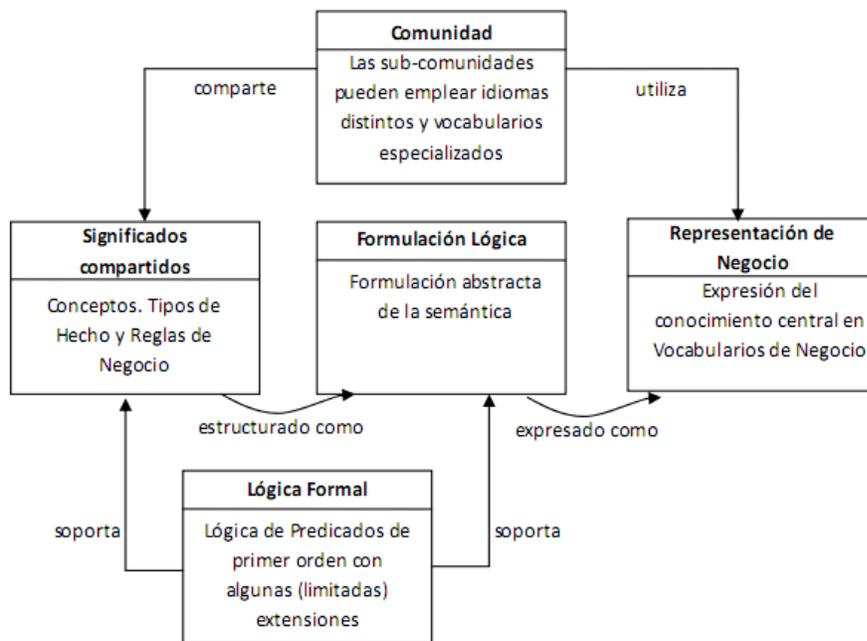


Figura 1.6: Principales elementos del metamodelo SBVR (OMG, 2008)

El 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 predicado de primer orden (Ross, 2008).

El metamodelo SBVR resultante se dirige no a personal del negocio sino a ingenieros de software encargados de construir herramientas para las personas del negocio. Puede incluirse o extenderse a modelos que consideran diferentes dominios de negocio. Proporciona los requisitos

⁶ Comunidades son la base del vocabulario de negocio, las de mayor importancia son las empresas para las que se están estableciendo y expresando las reglas de negocio.

básicos de integración y trazabilidad, y es consistente con los objetivos de la comunidad de representación del conocimiento (BRG, 2009).

1.4 Validación de la consistencia de reglas de negocio

La validación de la consistencia y los conflictos es una de las tareas más delicadas y consumidoras de tiempo en las investigaciones de RN, tanto como la semántica de las Reglas, especificada en el lenguaje natural, que resultan esenciales para comprender y desarrollar SI (Bajec and Marjan, 2006, Youdeowei, 1997). En las RN cada palabra tiene que convenir un significado preciso, definido en el vocabulario de negocio. Es el objeto de estudio de la presente investigación.

En el descubrimiento de las RN la organización es dividida en unidades manejables. No obstante, diferentes áreas de la organización tienen que responder al mismo tipo de eventos de negocio y tienen información similar, necesaria para completar estos eventos; estas áreas usualmente no comparten el mismo vocabulario, procedimientos y métodos para localizar los recursos que les permiten cumplir su misión (Bajec and Marjan, 2006, Moriarty, 2000). Es por esta razón que las reglas, desde diferentes unidades, muchas veces pueden ocasionar conflictos con otras y de esta manera ser inconsistentes. Para eliminar dichos conflictos e inconsistencias el analista del negocio debe realizar un estudio de las condiciones y las acciones que se plantean en las reglas. Las condiciones son las variables que se chequean en las reglas para realizar una determinada acción, son los posibles estados de una entidad, que el analista llama variables de decisión. Las acciones son opciones, que comprenden pasos, actividades o requerimientos, que puede elegir una persona cuando se enfrenta a un conjunto de condiciones. La validación de las RN se puede analizar como un proceso de toma de decisiones para eliminar los conflictos que se puedan presentar entre las reglas del repositorio.

La validación de conflictos e inconsistencia se hace inicialmente en el modelo de negocios, esto se repite cada vez que se adiciona una regla que se descubre en el desarrollo de un SI particular. Uno de los métodos que se están utilizando para realizar la validación de la consistencia de RN para evitar los conflictos y las inconsistencias en el repositorio de reglas es el empleo de tablas de decisión, árboles de decisión y español o inglés estructurado (Vanthienen, 2008).

1.4.1 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. No hay requerimientos de que el árbol tenga que ser simétrico, por lo que solamente se encuentran en una rama particular aquellas condiciones y acciones que son críticas para las decisiones presentes.

Cuando se dibujan los árboles de decisiones es útil distinguir entre las condiciones y las acciones, para este propósito, se utiliza un nodo cuadrado para indicar una acción y un círculo para una condición, tal y como se muestra en la figura 1.8.

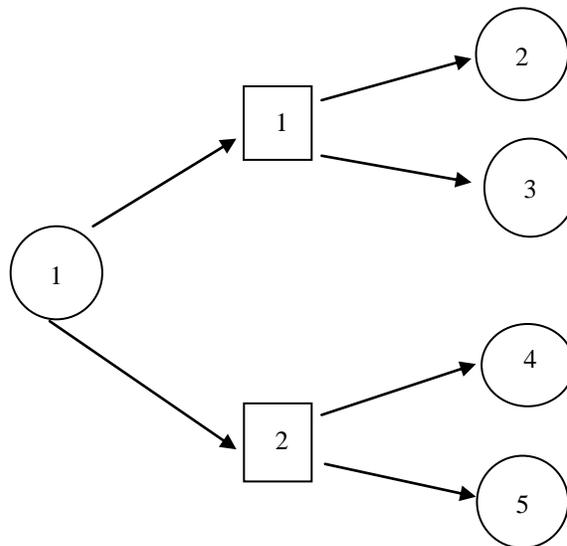


Figura 1.8: Árboles de decisión.

1.4.1.1 Características y ventajas de los árboles de decisión

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.

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. La parte que se encuentra a la derecha del árbol indica las acciones que deben realizarse, las que a su vez dependen de la secuencia de condiciones que las proceden (Vaniachine et al., 2005, Vanthienen et al., 2009).

El desarrollo de árboles de decisión beneficia al analista en dos formas. Primero que todo, la necesidad de describir condiciones y acciones llevan a los analistas a identificar de manera formal las decisiones que actualmente deben tomarse. De esta forma, es difícil para ellos pasar por alto cualquier etapa del proceso de decisión, sin importar que este dependa de variables cualitativas o cuantitativas (Vanthienen, 2008, Mass Chaviano and Marquez Garcia, 2010).

Los árboles de decisión también obligan a los analistas a considerar la secuencia de las decisiones.

El árbol de decisiones tiene tres ventajas principales sobre las tablas de decisiones, se muestran a continuación (Schewe, 2001, Ross, 2008):

1. **Primera:** Es que 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.
2. **Segundo:** 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.
3. **Tercero:** Al compararse con las tablas, los árboles de decisiones se entienden con más facilidad en una organización y son apropiados como un método de comunicación.

1.4.1.2 Identificación de los requerimientos

Los árboles de decisión también son útiles para identificar los requerimientos de datos críticos que rodean al proceso de decisión; es decir, los árboles indican los conjuntos de datos que la gerencia requiere para formular decisiones o tomar acciones.

Los árboles de decisiones se construyen después de completar el análisis de flujo de datos, entonces es posible que los datos críticos se encuentren ya definidos en el diccionario de datos. Si únicamente se utilizan árboles de decisión, entonces el analista debe tener la certeza de identificar con precisión cada dato necesario para tomar la decisión (Rosca et al., 1997a).

Los analistas necesitan describir y definir todos los datos utilizados en la toma de decisiones para que sea posible diseñar el sistema de forma tal que los genere apropiadamente.

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 que políticas o estrategias de la empresa son la guía para la toma de decisiones específicas (Goedertier and Vanthienen, 2006a).

Otro método para el análisis de decisiones es el español estructurado que se explica en el siguiente epígrafe.

1.4.2 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.

Aún 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.

1.4.2.1 Declaraciones estructuradas

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.

Cada uno de los métodos de análisis de decisiones existentes tiene sus ventajas, se utilizan de acuerdo con ellas. Las tablas de decisión proporcionan análisis completo de situaciones complejas y a la vez limitan la necesidad por cambios atribuibles a situaciones imposibles, redundancias o contradicciones. Los árboles de decisiones son importantes cuando es crítica la secuencia adecuada de condiciones y acciones, y cuando cada condición no es relevante para cada acción.

La validación que se quiere realizar en esta investigación es sobre el repositorio de RN desde el punto de vista de la semántica de las reglas, de su significado. Las RN están referidas al trasplante renal, los aspectos que tienen que tener en cuenta los trabajadores de salud para llevar a cabo dicha labor tan humana y meritoria de la cual depende la vida de los pacientes que se encuentren con *insuficiencia renal crónica*.

1.4.3 Tablas de decisión

Las tablas de decisión es una herramienta que puede utilizarse en los distintos momentos de un proyecto, esto es: en la exposición de los hechos, en el análisis del sistema actual, en el diseño del nuevo sistema y en el desarrollo del software (Linehan and De Sainte Marie, 2011, Castilla, 2007, Chang, 2004) para examinar, describir y documentar decisiones. Se emplean desde mediados de la década de los cincuenta, cuando fueron desarrolladas por la General Electric para el análisis de funciones de la empresa como control de inventarios, análisis de ventas, análisis de créditos y control de transporte y rutas (Vanthienen, 2003).

1.4.3.1 Definición de Tabla de Decisión como estructura algebraica

Una tabla de decisión puede ser definida como una tripla $D = (C, A, R)$ donde “C” es un conjunto de condiciones, “A” es un conjunto de acciones, y “R” es un conjunto de reglas.

Condiciones

Sea $C = [C(1), C(2), C(3), \dots]$ un conjunto de condiciones $C(1), C(2), C(3), \dots$. Cada condición $C(i)$ consiste de un sujeto $CS(i)$ y un conjunto de alternativas $[CA(i, j), j=1, 2, \dots, m(i)]$ y es escrito como $C(i): CS(i), [CA(i, 1), CA(i, 2), \dots]$.

Espacio de condiciones C'

$C' = C(1) \times C(2) \times C(3) \times \dots \times C(n)$ donde “x” denota el producto cartesiano, “C” es una colección de puntos tales que es definido como una “n-tupla” ordenada

$(CA(1, K1), CA(2, K2), \dots)$ donde $CA(1, K1) \in [CA(1, 1), CA(1, 2), \dots]$ y $CA(2, K2) \in [CA(2, 1), CA(2, 2), \dots]$, etc.

Acciones

Sea $A = [A(1), A(2), A(3), \dots]$ un conjunto de acciones donde cada acción $A(i)$ consiste de un sujeto $AS(i)$ y un conjunto de alternativas $[AA(i, j), j=1, 2, \dots, m(i)]$ escrita como:

$A(i): AS(i), [AA(i, 1), AA(i, 2), \dots]$.

Espacio de acciones A'

$A' = A(1) \times A(2) \times A(3) \times \dots \times A(n)$ donde “x” denota el producto cartesiano, “A” es una colección de puntos que es definido como una “n-tupla” ordenada

$(AA(1, K1), AA(2, K2), \dots)$ donde $AA(1, K1) \in [AA(1, 1), AA(1, 2), \dots]$ y $AA(2, K2) \in [AA(2, 1), AA(2, 2), \dots]$, etc.

Relación R

La relación “R” mapea puntos de C' dentro de puntos de A' , esto consiste en una serie de correspondencias o reglas que asocia un punto de C' con un punto en A' .

1.4.3.2 Estructura de una Tabla de Decisiones

Las tablas de decisiones están estructuradas en cuatro secciones: identificación de condiciones, entradas de condiciones, identificación de acciones y entradas de acciones (ver figura 1.7). La identificación de condiciones señala aquellas condiciones que son relevantes. Las entradas de condiciones indican qué valor, si es que lo hay, se debe asociar para una determinada condición e identificar las alternativas de decisión posibles (tales como S o N) para cada condición. La identificación de acciones enlista el conjunto de todos los pasos que se deben seguir cuando se presenta cierta condición. Las entradas de acciones muestran las acciones específicas del conjunto que deben emprenderse cuando ciertas condiciones o combinaciones de estas son verdaderas (Ross, 2006). En ocasiones se añaden notas en la parte inferior de la tabla para indicar cuando utilizar la tabla o para diferenciarla de otras tablas de decisión (Castilla, 2007).

Las columnas del lado derecho de la tabla enlazan condiciones y acciones, forman reglas de decisión que establecen condiciones que deben satisfacerse para emprender un determinado conjunto de acciones. La regla de decisión incorpora todas las condiciones que deben ser ciertas y no solo una a la vez.

(1) Identificación de Condiciones	(2) Entradas de condiciones.
(3) Identificación de Acciones	(4) Entradas de Acciones.

Figura 1.7: Esquema de Tablas de Decisión

Las tablas de decisión representan un conjunto completo de expresiones condicionales mutuamente excluyentes en un área específica.

1.4.3.3 Tipos de reglas y equivalencias

En cada regla se distinguen dos partes, la primera que corresponde a las condiciones y la segunda a las acciones, separadas por una doble línea. En la parte de las condiciones se pueden colocar tres tipos de entradas: **S** (si), **N** (no) y – (indiferencia) (Vanthienen et al., 2009).

La entrada **S** (si) significa que la condición debe satisfacerse o que es cierto que la condición se satisface. La entrada **N** (no) significa que la condición no debe cumplirse o no se cumple. El – (indiferencia) significa que no importa que la condición se cumpla o no.

En la parte de reglas de las acciones hay dos tipos de entradas: **X** se debe realizar esa acción y el – que no se debe realizar esa acción.

A las reglas que en la parte de las condiciones poseen únicamente entradas **S** y/o **N** se les llama reglas puras; a las reglas que en la parte de condiciones poseen por lo menos una entrada – (indiferencia), se les llama reglas mixtas.

El número posible de reglas puras en una tabla de decisión es igual a 2 elevado al número de condiciones que posea la tabla, donde el 2 representa las dos posibles respuestas: S o N.

Equivalencia entre reglas puras y mixtas

Toda regla mixta puede descomponerse en dos o más reglas puras (Ross, 2008).

Consideremos la siguiente regla mixta:

Mixta		Pura		Pura
S		S		S
N		N		N
-	=	S	+	N
N		N		N
X		X		X
-		-		-
X		X		X

La regla mixta anterior puede descomponerse en dos reglas puras. En aquella *Condición* donde hay una indiferencia se puede tener S o N.

Si tengo la siguiente regla mixta:

Mixta		Pura		Pura		Pura		Pura	
S		S		S		S		S	
-		S		S		N		N	
N	=	N	+	N	+	N	+	N	
-		S		N		S		N	
X		X		X		X		X	
-		-		-		-		-	
X		X		X		X		X	
		S				S			
		S				N			
		N				N			
		-				-			
		=				+			
		X				X			
		-				-			
		X				X			

La regla mixta tiene dos indiferencias en las condiciones y se pudo descomponer en cuatro reglas puras.

Cuando se tenía una indiferencia se pudo desdoblar la regla mixta en dos puras.

Cuando se tenían dos indiferencias se pudo desdoblar la regla mixta en cuatro puras, por lo tanto generalizando:

Siendo N el número de indiferencias que contiene una regla mixta, el número total de reglas puras contenidas en ella es 2^N .

Existen varios tipos de anomalías que se pueden identificar con tablas de decisión, y se explicarán en el siguiente epígrafe.

1.4.3.4 Tipos de anomalías identificables

En el repositorio de RN se pueden identificar varios tipos de anomalías, analizando las reglas desde el punto de vista lógico a través de tablas de decisión (Vanthienen et al., 2009). Cuando se clasifican posibles anomalías en tablas se pueden distinguir entre casos de redundancia, ambivalencia, circularidad y deficiencia (O'Keefe and O'Leary, 1993, Preece and Shinghal, 1994). De acuerdo al lugar donde ocurren, se pueden distinguir dos tipos de anomalías: intra-tabular e inter-tabular. La anomalía intra-tabular se puede ver claramente que ocurre entre los componentes dentro de una misma tabla simple, 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.

Debido a que estos tipos de anomalías, que se pueden detectar con las tablas de decisión, hacen que el repositorio de RN sea inconsistente, es que se pasa a la depuración del mismo.

1.4.3.5 Depuración de las tablas de decisión

El proceso de depuración de las tablas de decisión se realiza con el objetivo de hacer más compacta la tabla, es decir que tenga menos reglas; encontrar posibles errores en las reglas por haber partido por ejemplo de un enunciado contradictorio, eliminación de redundancias para hacer más consistente el repositorio de R. Se puede aplicar sin hacer referencia a los contenidos del enunciado, simplemente trabajando con la estructura que presentan las reglas y una vez finalizado el proceso volver a reemplazar por los enunciados de las condiciones y las acciones. Se realiza a través de tres leyes y un criterio.

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 presenta 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 los 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, con iguales acciones, 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 se ha construido mal la tabla por lo que hay que investigar el error antes de continuar.

Segunda Ley: Eliminación de redundancias

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:

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.

Las tablas de decisión han sido estudiadas o aplicadas en muchos contextos de investigación, tales como la verificación y validación de sistemas basada en el conocimiento, la ejecución eficiente de los sistemas basados en el conocimiento, mantenimiento base de conocimientos, la adquisición de conocimientos, el descubrimiento de conocimiento, varios dominios de aplicación, tales como medicina y derecho. Un nuevo enfoque es el uso de tablas de decisión para la gestión de RN. Las RN se han convertido en un enfoque común para el entendimiento, especificación e implementación de procesos de negocios (Linehan and De Sainte Marie, 2011). Este nuevo enfoque se suma al trabajo que se ha hecho en las tablas de decisión como una tecnología basada en el conocimiento de sistemas (Valiente, 1991, Vanthienen, 2008). Y, la técnica de tabla de decisión está siendo continuamente usado y vuelto a descubrir en la representación y la validación de la lógica de negocios complejos en las RN. Un ejemplo de software que se inspira en este enfoque es el PROLOGA (Analizador de lógica de PROcedimiento), que es un conjunto de métodos y herramientas para la especificación,

validación e implementación de sistemas de información, por medio de tablas de decisión (Vanthienen, 2003).

1.5 Conclusiones parciales del capítulo

Durante la confección de este capítulo, como búsqueda bibliográfica teórica del tema de validación del conjunto de RN, se expusieron las diferentes etapas por las que transcurren los sistemas basados en el enfoque de RN. Se realizó un análisis de la metodología PROTEUS[®] y las diferentes vías por las cuales se puede realizar la validación de la consistencia de RN y se llegaron a las siguientes conclusiones.

1. Ubicación de la validación de la consistencia del conjunto de RN en las etapas de análisis de reglas de negocio y de las reglas del producto o servicio, en los pasos 6 y 11 respectivamente dentro de la metodología PROTEUS[®].
2. Se identifican varios métodos para la validación, centrandolo en el presente trabajo en tablas de decisión, delimitando el alcance del trabajo.
3. Se detallan los tipos de anomalías que pueden ser detectadas en el conjunto de reglas de negocio a partir de las tablas de decisión.

Capítulo II: Métodos de validación de la consistencia sobre tablas de decisión

En este capítulo se realiza por parte del autor el análisis de diferentes métodos de validación de la consistencia, primero trata la validación basada en patrones de diseño de las RN y luego otros basados en tablas de decisión.

No existe un estándar de cómo hacer RN, 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.

2.1 Patrones de reglas de negocio

Los patrones de reglas son recursos semánticos usados para describir las RN. Las reglas restringen la semántica de los conceptos y de las relaciones conceptuales de un dominio de aplicación particular. A través del modelo de hechos con enfoque ontológico se especifica, de forma explícita, una conceptualización de un dominio que puede llegar a ser consensuado; donde los patrones representan la sintaxis y permiten describir las reglas.

La sintaxis de dichos patrones se define utilizando conceptos del vocabulario del negocio manejados en el modelo de hechos, descrito en el capítulo anterior (Martínez Busto et al., 2009, Núñez Pérez et al., 2010).

Con el apoyo semántico que brinda el modelo de hechos genérico con enfoque ontológico, descrito anteriormente, se definen un conjunto de patrones que sirven como estructura para la descripción de las RN. Se utiliza para ello la taxonomía de clasificación semántica de las reglas propuesta por (Weiden et al., 2004). De los tres niveles de expresión para las RN (Morgan, 2002): informal, técnico y formal; son los dos primeros los de interés de esta investigación.

Los patrones son una sentencia o afirmación y se elaboran sobre la versión de una estructura básica, por ejemplo:

<sujeito> debe <restricción>

El conjunto de patrones propuesto puede ser usado en la modelación de las RN para un dominio específico, teniendo como base conceptos y definiciones planteados esencialmente por Weiden (2004) y Morgan (2002) debido a la claridad que ofrecen para la comprensión de las reglas. Algunas de las estructuras utilizadas para la definición de esos patrones son:

Sujeto: entidad reconocible del negocio, o sea, objetos del negocio visibles en el modelo de hechos.

Característica: comportamiento del negocio que debe llevarse a cabo o una interrelación que debe ser establecida; en fin, es la condición que debe cumplir el sujeto.

Se han obtenido diecisiete patrones, asociados a las diferentes subcategorías definidas para la clasificación semántica, aquí solamente hacemos referencia a algunos de ellos.

Seguidamente a modo de ejemplo se muestra la sintaxis del patrón 1: estructura de concepto, para una Regla Estructural.

Patrón 1:

Restricción Básica: Es el más común de los patrones de RN, establece una restricción sobre un sujeto de una regla.

<determinante> <sujeto> [no] (debe |tiene) <característica>

[(si | a menos que) <hecho>].

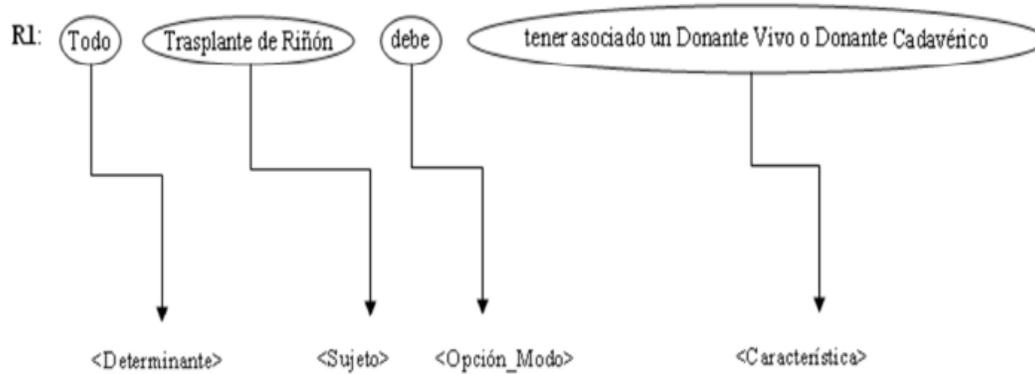
< determinante > < sujeto > (puede < característica > solo si <hecho>) |

(no puede < característica >).

Ejemplos:

1. Cada paciente con IRCT tiene que ser valorado en Consulta de Nefrología.
2. Un *Posible Receptor* puede ser incluido en la *Lista de Espera de Trasplante* solamente si fue valorado completamente por la *Comisión de Trasplante*.

3. Todo Trasplante de Riñón debe tener asociado un Donante Vivo o Donante Cadavérico.



Patrón 2:

Lista de Restricción: este patrón también restringe al sujeto, pero restringiendo que son uno o más ítems tomados de una lista.

< determinante > < sujeto > [no] (debe | tiene) < característica >

(si | a menos que) como mínimo <m>

[no más de <n>] de las siguientes es verdadera:

<lista de hechos>.

< determinante > < sujeto > (puede < característica > solo si) |

(no puede < característica > si)

al menos <m> [y no más de <n>] de las siguientes es verdadera:

<lista de hechos>.

Ejemplos:

1. El Grupo de Alto Riesgo tiene que ser integrado por Pacientes que cumplan una de las siguientes características:
 - a. Un paciente con serología que releve hepatitis C o B.
 - b. Un paciente con evidencias de actividades bioquímicas.

- c. Un paciente con evidencia de actividades histológicas.
2. Un Donante Potencial no puede ser un Donante si presenta cualquiera de las siguientes Contraindicaciones:
 - a. Tiene VIH,
 - b. Hepatitis B,
 - c. Hepatitis C

Patrón 3:

Clasificación: este patrón establece una definición para un término en el modelo de hechos. Tales términos son equivalentes a términos de corta vida o de uso temporal. Si esta clasificación es permanente se debe reflejar mejor en el Modelo de Hechos.

< determinante > < sujeto > [no] es definido como <clasificación>

[(si | a menos que) <hecho>].

< determinante > < sujeto > [no] debe ser considerado como < clasificación >

[(si | a menos que) < hecho >]

Ejemplo:

1. Una *Operación de Trasplante* es definida como un *Retrasplante* si al Receptor se le realizó *al menos* una *Operación de Trasplante*.

Patrón 4:

Cálculo: este patrón establece una interacción entre términos en el Modelo de Hechos suficientes para realizar los cálculos o el establecer un valor. Note que esto es similar al patrón de clasificación, en cuanto a obtención de valores temporales. Se usa tanto “es definido como” y “debe ser calculado como”.

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

< determinante > < resultado > = < algoritmo >.

Ejemplo:

1. La Hidratación de un Receptor en la Sala de Trasplante = Peso del Receptor en Kg * (30 or 35 cc).

Patrón 5:

Enumeración: este patrón establece el rango de valores que pueden ser legalmente tomados por un término en el Modelo de Hechos.

< determinante > < resultado > debe ser elegido desde la siguiente lista

[abierta | cerrada]: <lista de valores>

Ejemplo:

1. El Motivo de la Pérdida de un Trasplante Renal debe ser seleccionado de la siguiente lista:
 - a. Causas médicas
 - b. Complicaciones quirúrgicas

Otro ejemplo es el patrón 7: Pre-Condición, para una regla de conducta con mayor complejidad:

Patrón 7:

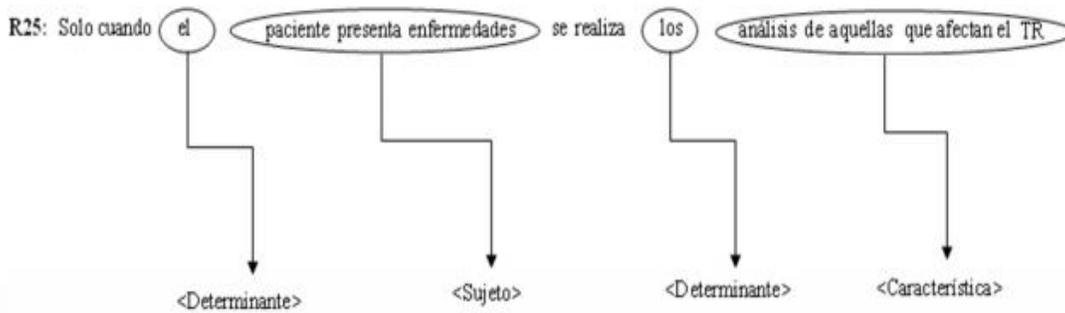
Solo cuando < determinante > < sujeto > se realiza

< determinante > < característica >

< sujeto > ::= < tarea > | <proceso > | <hecho >

<carácterística > ::= < tarea > | <proceso > | <hecho >

Un ejemplo de RN para este patrón es el siguiente:



Dichos patrones permiten establecer un lenguaje español estructurado (subconjunto del español con estructuras que lo hacen rígido), necesario para captar RN mediante una herramienta de software, siguiendo el principio de que dicho lenguaje sea cercano a los usuarios del negocio. Las herramientas de edición de las RN 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.

Seguidamente explicaremos como construir las tablas de decisión en las que se basa otro método de validación de RN.

2.2 Tabla de decisión para la validación del conjunto de reglas

Según la bibliografía consultada una tabla de decisión no es más que una herramienta que sintetiza procesos en los cuales se dan un conjunto de condiciones y un conjunto posible de acciones a realizar en dependencia del valor que toman dichas condiciones. Su estructura permite el análisis para la toma de decisiones ya que muestra todas las posibles combinaciones de valores de las condiciones, esta es una de las principales aplicaciones de las mismas. *María Josefina Castilla* es una destacada investigadora de la *Universidad de Murcia, España*, que ha tratado sobre el tema, en sus diferentes aplicaciones como en el desarrollo de sistemas de información y en la toma de decisiones. *Jan Vanthienen*, es otro investigador destacado de la *Universidad Católica de Leuven, Bélgica*, ha desarrollado diferentes investigaciones acerca de tablas de decisión como validación de la consistencia de RN y procesos.

2.2.1 Construcción de las tablas de decisión

Las tablas de decisión son de gran utilidad debido a todas las facilidades que brindan (Linehan and De Sainte Marie, 2011, Castilla, 2007):

- 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 decisión estructuradas.
- Ayuda a la toma de decisiones.

Como se explicó en el Capítulo I las tablas de decisión está integrada por: matriz de condiciones, matriz de condiciones y matriz de reglas para condiciones y acciones. En la matriz de condiciones se enumeran todas las situaciones que puedan 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. 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. 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 multiplica el número de alternativas de cada condición. Si hay cuatro condiciones y cada una tiene dos alternativas (S y N) existirán dieciséis posibilidades, según se indica:

Condición 1:	x	2 alternativas
Condición 2:	x	2 alternativas
Condición 3:	x	2 alternativas
Condición 4:	x	2 alternativas
		16 posibilidades

- 5) Rellenar las alternativas de las distintas condiciones, existen dos formas de hacerlo. Se comienza con la primera condición y se divide el número de columnas por el número de alternativas de cada condición. En el ejemplo precedente hay dieciséis columnas y dos alternativas (S y N), por lo tanto dieciséis dividido por dos nos da ocho. A continuación se escogen una de las alternativas, por ejemplo S, y se escribe S en las ocho columnas. Se escoge luego la otra alternativa, la N, y se escribe N en las restante ocho columnas (Goedertier et al., 2007b):

Condición 1 **SSSSSSSSNNNNNNNN**

Se repite el mismo procedimiento con cada condición utilizando un subconjunto de la tabla:

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

y se continúa completando la tabla con cada condición.

Condición 1	S	S	S	S	S	S	S	S	N	N	N	N	N	N	N	N
Condición 2	S	S	S	S	N	N	N	N	S	S	S	S	N	N	N	N
Condición 3	S	S	N	N	S	S	N	N	S	S	N	N	S	S	N	N
Condición 4	S	N	S	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.
- 7) La segunda forma de rellenar la tabla considera una condición a la vez y, por cada condición adicional le añade una tabla pero sin considerar las combinaciones de condiciones y acciones duplicadas.

Veamos con un ejemplo como se va construyendo la tabla de decisión en el siguiente epígrafe.

2.2.2 Construcción de la tabla de decisión

Descripción del ejemplo:

Cuando un cliente de una Empresa paga dentro de los 30 días y la cantidad solicitada no supera el stock, se factura con descuento y se envía la mercadería solicitada. Sin embargo si el pago se hiciera después de los 30 días se facturaría sin descuento, remitiendo la mercadería. Las mismas acciones se emprenden si se tratara de un cliente

nuevo. Hacer lo mismo cualquiera sea el plazo de pago. Si no existe cantidad suficiente en el stock y se trata de un cliente de la empresa que paga dentro de los 30 días, facturar con descuento, realizando la entrega de la cantidad en el stock y dejar pendiente el resto del pedido. Si el cliente fuera nuevo no realizar descuento alguno (Contreras Baez and Bolanos Rodríguez, 2011).

En caso de que el pago no se efectuara dentro de los 30 días, cualquiera sea el cliente, se procedería de esta última manera. Si un cliente que compra por primera vez, solicita mayor cantidad de mercadería que la de stock, cualquiera sea el plazo de pago, no se le practicará descuento alguno, remitiéndose la cantidad en stock y dejando pendiente la diferencia (Ross, 2008).

Separamos las condiciones y las acciones:

Condiciones	Acciones
Cliente de la empresa	Facturar con descuento
Paga dentro de los 30 días	Enviar mercadería solicitada
Cantidad solicitada no supera el stock	Facturar sin descuento
Pagó después de los 30 días	Dejar pendiente lo solicitado menos el stock
Cliente Nuevo Cantidad no supera el stock Cualquiera sea el plazo No hay suficiente cantidad en stock Cliente por primera vez Mayor cantidad de mercadería que la de stock	Enviar stock

Se normaliza el lenguaje y se construye la tabla.

	1	2	3	4	5	6	7	8
Cliente de la empresa	S	S	N	N	S	N	-	N
Plazo de pago \leq 30 días	S	N	N	-	S	S	N	-
Cantidad solicitada \leq cantidad en stock	S	S	S	S	N	N	N	N
Facturar con descuento	X	-	-	-	X	-	-	-
Facturar sin descuento	-	X	X	X	-	X	X	X
Enviar mercadería solicitada	X	X	X	X	-	-	-	-
Enviar cantidad en stock	-	-	-	-	X	X	X	X
Dejar pendiente cantidad solicitada-stock	-	-	-	-	X	X	X	X

Figura 2.1: Construcción de tabla de decisión

En cada regla se distinguen dos partes, una que corresponde a las condiciones y otra a las acciones, separadas por una doble línea (ver figura 2.1). En la parte de las condiciones se pueden colocar tres tipos de entradas: S (si), N (no), - (indiferencia).

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

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

El - (indiferencia) significa que no importa que la condición se cumpla o no.

En la parte de reglas de las acciones hay dos tipos de entradas: X se debe realizar esa acción y el - que no se debe realizar esa acción.

A las reglas que en la parte de las condiciones poseen únicamente entradas S y/o N se les llama reglas puras; a las reglas que en la parte de las condiciones poseen por lo menos una entrada - (indiferencia), se les llama reglas mixtas.

2.2.3 Eliminación de inconsistencias y ambigüedades en el conjunto de reglas

Mediante este proceso se logra 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.

Este proceso puede aplicarse sin hacer referencia a los contenidos del enunciado, simplemente trabajando con la estructura que presentan las tablas y una vez finalizado el proceso volver a reemplazar por los enunciados de las condiciones y de las acciones. Se realiza a través de tres leyes y un criterio que se enunciaron en el Capítulo I.

Aplicamos en estos momentos la primera ley *Unificación de reglas por indiferencia*, agrupando las reglas en todos los pares posibles para analizarlos y diferenciar los pares 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, con el objetivo de eliminar a ambas de la tabla y reemplazarlas por una regla con idénticas entradas de condiciones y acciones, que registren la entrada indiferente

(-) en aquella condición donde las originales diferían.

En nuestro ejemplo las reglas 2 y 3 tienen idénticas entradas de condiciones y acciones, salvo en la condición 1 (Cliente de la empresa) donde una registra una entrada S y la otra una N, entonces las podemos eliminar, reemplazándola por una regla mixta. La tabla queda como se muestra en la figura 2.2:

	1'	2'	3'	4'	5'	6'	7'
Cliente de la empresa	S	-	N	S	N	-	N
Plazo de pago \leq 30 días	S	N	-	S	S	N	-
Cantidad solicitada \leq cantidad en stock	S	S	S	N	N	N	N
Facturar con descuento	X	-	-	X	-	-	-
Facturar sin descuento	-	X	X	-	X	X	X
Enviar mercadería solicitada	X	X	X	-	-	-	-
Enviar cantidad en stock	-	-	-	X	X	X	X
Dejar pendiente cantidad solicitada-stock	-	-	-	X	X	X	X

Figura 2.2: Tabla después de aplicada una reducción

Una vez aplicada la primera ley, se agrupan las reglas nuevamente por pares y se analizan los que resulten diferenciados por al menos una condición, se le aplica un criterio para clasificarlas en *reglas independientes y dependientes* teniendo en cuenta si en ese 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 son *independientes*, el par de reglas que no cumpla con este requisito se dice dependiente. Si las reglas son dependientes es posible determinar si son redundantes o contradictorias.

Como se explicó en el Capítulo I un par de reglas dependientes son redundantes si las reglas tienen las mismas entradas de acciones, es decir, cuando solamente en las condiciones difieren en una sola entrada de una condición donde una registra una entrada S y la otra una entrada N y tienen las mismas acciones a realizar. Se dice que un par de reglas dependientes son contradictorias cuando bajo solamente una diferencia en el valor de entrada de una condición, expresan diferentes entradas de acciones.

En el caso de que el par analizado sean *dependientes redundantes* puede aplicarse la segunda y tercera ley, sino significa que hay un error en el enunciado o que se ha construido la tabla por lo que hay que investigar el error antes de continuar.

En este ejemplo como el resultado de aplicar la primera ley y el criterio resulta lo siguiente:

1' y 2': son independientes	3' y 4': son independientes
1' y 3': son independientes	3' y 5': son independientes
1' y 4': son independientes	3' y 6': son independientes
1' y 5': son independientes	3' y 7': son independientes
1' y 6': son independientes	4' y 5': son independientes
1' y 7': son independientes	4' y 6': son independientes
2' y 3': son dependientes redundantes	4' y 7': son independientes
2' y 4': son independientes	5' y 6': son independientes
2' y 5': son independientes	5' y 7': son dependientes redundantes
2' y 6': son independientes	6' y 7': son dependientes redundantes
2' y 7': son independientes	

En este paso se comienzan a analizar los pares que resultaron dependientes redundantes para tratar de eliminar la redundancia aplicando la segunda y tercera ley, explicadas en el Capítulo I.

Segunda Ley: 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.

Consideremos en nuestro ejemplo los pares de reglas que resultaron dependientes y redundantes:

2' y 3': son ambas mixtas.

6' y 7': son ambas mixtas.

5' y 7': una es pura y la otra es mixta por lo que puede aplicarse la segunda ley.

Descomponiendo la mixta el par queda de la siguiente forma:

5'	7'a	7'b
N	N	N
S	S	N
N	N	N

La regla 5' está contenida en la 7'a por lo tanto puede ser eliminada quedando la tabla como se muestra en la figura 2.3.

	1''	2''	3''	4''	5''	6''
C1	S	-	N	S	-	N
C2	S	N	-	S	N	-
C3	S	S	S	N	N	N
A1	X	-	-	X	-	-
A2	-	X	X	-	X	X
A3	X	X	X	-	-	-
A4	-	-	-	X	X	X
A5	-	-	-	X	X	X

Figura 2.3: Tabla de decisión resultante de aplicar la segunda ley para reducir redundancias

Para eliminar la redundancia en los pares donde ambas reglas sean mixtas se aplica la tercera ley explicada en el Capítulo I que plantea lo siguiente “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”. En nuestro ejemplo existen dos pares de reglas que cumplen con esta condición 2'' y 3'' y 5'' y 6'', que son dependientes redundantes y mixtas. Por lo tanto aplicamos esta tercera ley a estas, realizando el siguiente procedimiento.

Descomponemos las reglas mixtas en las puras correspondientes:

2''a	2''b	3''a	3''b	5''a	5''b	6''a	6''b
S	N	N	N	S	N	N	N
N	N	S	N	N	N	S	N
S	S	S	S	N	N	N	N

Podemos eliminar la regla 2''b y en el otro par la regla 5''b, la tabla queda:

	1''	2''	3''	4''	5''	6''
C1	S	S	N	S	S	N
C2	S	N	-	S	N	-
C3	S	S	S	N	N	N
A1	X	-	-	X	-	-
A2	-	X	X	-	X	X
A3	X	X	X	-	-	-
A4	-	-	-	X	X	X
A5	-	-	-	X	X	X

Figura 2.4: Tabla de decisión resultante de aplicar la tercera ley para reducir redundancias

En nuestro ejemplo ya todas las reglas son independientes, sustituyendo en la tabla las variables C1, C2 y C3 por las respectivas condiciones reales y A1, A2, A3, A4 y A5 por las acciones correspondientes, queda la tabla como muestra la figura 2.5.

	1"	2"	3"	4"	5"	6"
Cliente de la empresa	S	S	N	S	S	N
Plazo de pago \leq 30 días	S	N	-	S	N	-
Cantidad solicitada \leq stock	S	S	S	N	N	N
Facturar con descuento	X	-	-	X	-	-
Facturar sin descuento	-	X	X	-	X	X
Enviar la mercadería	X	X	X	-	-	-
Enviar el stock	-	-	-	X	X	X
Dejar pendiente cantidad solicitada - stock	-	-	-	X	X	X

Figura 2.5: Tabla de decisión resultante libre de redundancias ni ambigüedades

No todas las tablas de decisión tienen como valor de entrada de las condiciones S (si), N (no) o - (indiferencia) y para las acciones X y - . Seguidamente mostraremos otras posibles estructuras de tablas de decisión existen.

2.2.4 Otros tipos de tablas de decisión

Los tipos de tablas de decisión que existen en la literatura son:

- Tablas de decisión de entradas limitadas.
- Tablas de decisión de entradas ampliadas.
- Tablas de decisión de entradas mixtas.

Todas las tablas de decisión donde las condiciones solamente tienen los valores: S, N y - , y las acciones X y - , son tablas de decisión de *entradas limitadas*. Cuando se construyen tablas con ellas el enunciado de cada condición está completo en la matriz de condiciones (Vanthienen, 2008).

En las tablas con *entradas ampliadas* son aquellas donde los enunciados de las condiciones no quedan completos en la matriz de condiciones sino que se terminan de completar en las reglas en la parte de condiciones. Los enunciados de acciones tampoco

se completan en la matriz de acciones sino que terminan de hacerlo en las reglas en la parte de las acciones. A continuación se muestra la tabla del ejemplo en forma ampliada.

	1	2	3	4	5	6
Cliente	De la Empresa	De la Empresa	De la Empresa	De la Empresa	Nuevo	Nuevo
Plazo de Pago	Menor o igual que 30 días	Menor o igual que 30 días	Mayor que 30 días	Mayor que 30 días	Cualquiera	Cualquiera
Cantidad solicitada	Menor o igual que la de stock	Mayor que la de stock	Menor o igual que la de stock	Mayor que la de stock	Menor o igual que la de stock	Mayor que la de stock
Facturar	Con descuento	Con descuento	Sin descuento	Sin descuento	Sin descuento	Sin descuento
Entregar	Mercadería solicitada	Mercadería en stock	Mercadería solicitada	Mercadería en stock	Mercadería solicitada	Mercadería en stock
Pendiente	-	Cantidad solicitada-stock	-	Cantidad solicitada-stock	-	Cantidad solicitada-stock

Figura 2.6: Tabla de decisión de entradas ampliadas del ejemplo analizado

Las tablas con *entradas mixtas* son aquellas en las que se mezclan entradas limitadas y entradas ampliadas. Por ejemplo:

	1	2	3	4	5	6
Cliente de la empresa	S	S	S	S	N	N
Plazo de Pago	Menor o igual que 30 días	Menor o igual que 30 días	Mayor que 30 días	Mayor que 30 días	Cualquiera	Cualquiera

Cantidad solicitada \leq stock	S	N	S	N	S	N
Facturar	Con descuento	Con descuento	Sin descuento	Sin descuento	Sin descuento	Sin descuento
Entregar mercadería solicitada	X	-	X	-	X	-
Entregar stock	-	X	-	X	-	X
Pendiente	-	Cantidad solicitada-stock	-	Cantidad solicitada-stock	-	Cantidad solicitada-stock

Figura 2.7: Tabla de decisiones de entradas mixtas del ejemplo analizado.

2.3 Herramienta PROLOGA

PROLOGA es una herramienta para realizar análisis lógico basado en tablas de decisión realizado por un investigador y profesor distinguido, Jan Vathienen (Vanthienen et al., 2009). Realiza verificación y validación de las RN donde utiliza estas tablas de decisión como forma de representar todas las condiciones y acciones. Una ventaja que brindan las tablas de decisión es que las columnas tienen que ser mutuamente excluyentes, Vanthienen las llama single-hit-table (Vanthienen et al., 2009). En una single-hit-table cada posible combinación de estado de las condiciones puede estar presente solamente en una y solo una columna de la tabla. Este criterio de exclusividad es un factor clave en el proceso de verificación debido a que previene varios tipos de redundancia y ambivalencia. En problemas reales la utilización de una sola tabla para representar todo el conocimiento no es suficiente, uno debe ser capaz de estructurar las decisiones lógicas dentro de un conjunto de tablas interrelacionadas. Con este fin PROLOGA distingue entre dos tipos de subtablas:

- Subtabla de condiciones, la cual determina el estado de una condición.
- Subtabla de acciones, la cual brinda más detalles de la acción.

Las relaciones inter-tabulares pueden ser visualizadas por medio de un grafo dirigido (ver figura 2.8). Cada nodo representa una tabla de decisión. Los arcos corresponden a los enlaces entre tablas relacionadas como padre e hija, desde subtablas de condiciones hacia subtablas de acciones.

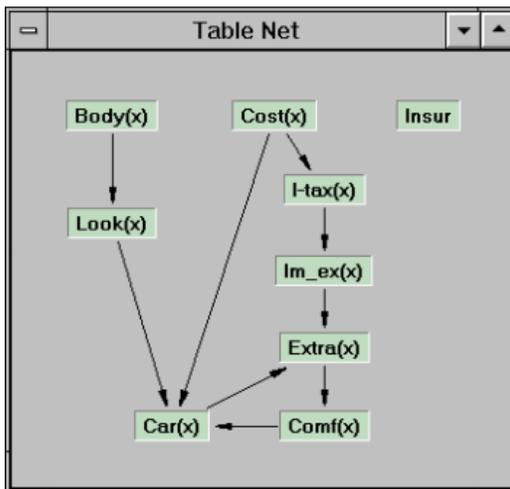


Figura 2.8: Estructura de un proyecto PROLOGA sobre tablas de decisión

El proceso de decisión de cómo modularizar el conocimiento puede ser hecho intuitivamente o automáticamente, dependiendo de la complejidad del caso.

2.3.1 Anomalías identificadas

Con la herramienta PROLOGA, se pueden detectar diferentes tipos de anomalías en sistemas basados en tablas de decisión (Vanthienen et al., 2009). Los tipos de anomalías que se pueden distinguir entre los explicados en el Capítulo I son: casos de redundancia, ambivalencia, circularidad y deficiencia (O'Keefe and O'Leary, 1993, Preece and Shinghal, 1994). Las distingue en función del lugar donde ocurren: intra-tabular e inter-tabular.

Anomalías inter-tabular:

- Redundancia

- Entrada de acción redundante
- Fila de acción inusual.
- Columna insatisfecha
- Ambivalencia
 - Entrada de acciones ambivalente
- Circularidad
- Deficiencia

Anomalías intra-tabular:

- Redundancia
 - En las columnas: pares de columnas duplicadas, condiciones insatisfechas.
 - En las filas: condiciones irrelevantes, par de filas redundantes.
- Ambivalencia
 - En el nivel de columna
 - En el nivel de fila

2.3.2 Anomalías del tipo inter-tabular

Son anomalías que ocurren entre tablas de un mismo problema en un proyecto de PROLOGA. Estas anomalías se visualizan en la herramienta a través de un grafo, es decir, una estructura de datos que representa las tablas existentes y sus relaciones. Asociado al ejemplo referido con anterioridad se muestra en la figura 2.9 dicho grafo en el cual se detecta la siguiente anomalía: a pesar de que intuitivamente la tabla Car(x) es la tabla principal, existe una flecha que apunta a la tabla Extra(x), esta tiene otra flecha que apunta a la tabla Comf(x) y esta a su vez tiene otra flecha que apunta a la subtabla Car(x) nuevamente, entonces se está en presencia de una cadena circular.

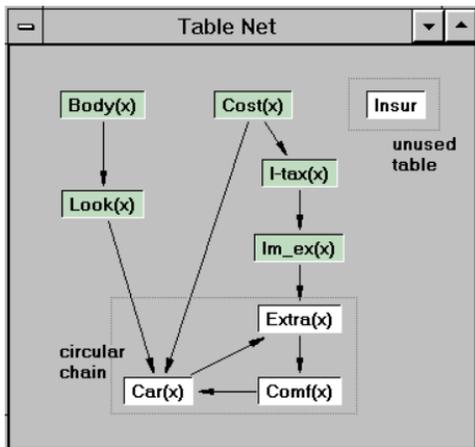


Figura 2.9: Anomalía inter-tabular de cadena circular

Además en el grafo se muestra una tabla de decisión aislada llamada $insur(x)$ que no tiene vínculo con ninguna otra tabla. En consecuencia a ello esta tabla sería un enlace perdido que no influiría en el proceso de decisión de tipos de carros.

El intercambio de las mismas condiciones o acciones entre diferentes tablas puede causar anomalías de redundancias o ambivalencias (Vanthienen et al., 1995).

2.3.3 Anomalías del tipo intra-tabular

Este tipo de anomalías ocurren entre los componentes dentro de una misma tabla de decisión, se explicará cómo PROLOGA lo hace. Los ejemplos están basados en casos que se pueden presentar en las tablas que se muestran en la figura 2.9.

Análisis de la tabla $Car(x)$

La tabla de decisión $Car(x)$ contiene los datos que se refieren a los diferentes tipos de vehículo, a saber, de lujo, económico o deportivo (Vanthienen et al., 2009). Se tienen las siguientes reglas tomadas de (Vanthienen et al., 2009):

- C38: $Luxury-car(x) \leftarrow look(x), comfortable(x), high_cost(x)$
- C40: $Soport-car(x) \leftarrow look(x), high_cost(x), speed(x), performance(x)$

PROLOGA detecta inmediatamente las reglas que se encuentran en conflicto. Las restricciones internamente están definidas por medio del operador "only". Este operador especifica que bajo una combinación de condiciones solamente se debe deducir una sola

conclusión. Tan pronto como es adicionada la segunda regla a la base de conocimientos se detecta la contradicción entre este par de reglas y se lanza una advertencia al usuario, tal como se muestra en la figura 2.10.



Figura 2.10: Mensaje de contradicción en la tabla Car(x)

Analizando la tabla Car(x) que resulta se detecta la anomalía al observar las columnas que en la parte de las acciones aparece el símbolo “?” (ver figura 2.11), es fácil deducir las que se encuentran en conflicto. Esto ilustra claramente como el formato de tabla de decisión puede ser una intuitiva y poderosa técnica de visualización. Brinda al analista una vía más fácil de comunicación con el experto de negocio, puede ser de ayuda sustancial durante la adquisición de conocimiento y mejorar la calidad de la base de conocimientos (Vanthienen et al., 2009).

1. $\hat{Cost}(x)$	High				Low	
2. $\hat{Look}(x)$	Y		N		-	
3. $\hat{Comfort}(x)$	Y	N		-	-	
4. High-mileage(x)	-	-	-	Y	N	
5. Speed(x)	Y	N	Y	N	-	-
6. Performance(x)	Y	N	-	Y	N	-
1. Car(x) := Luxury Car	?	x	x	-	.	.
2. Car(x) := Economical Car	-	-	-	-	.	x
3. Car(x) := Sport Car	?	-	-	x	.	.

Figura 2.11: Tabla de decisión Car(x) con anomalías

Además de la tabla de decisión anterior se puede deducir que es *incompleta* ya que existen columnas en que con valores de condiciones válidos no se especifica qué acción realizar, como es el caso de la última columna donde $Cost(x) = \text{“Low”}$, $High-mileage(x) = \text{“N”}$ y las demás condiciones no importa el valor que tomen, no se deduce ningún conocimiento, es decir, no se llega a clasificar el carro. Puede verse en la figura

2.12 el reporte de verificación intra-tabular que se puede consultar en cualquier momento.

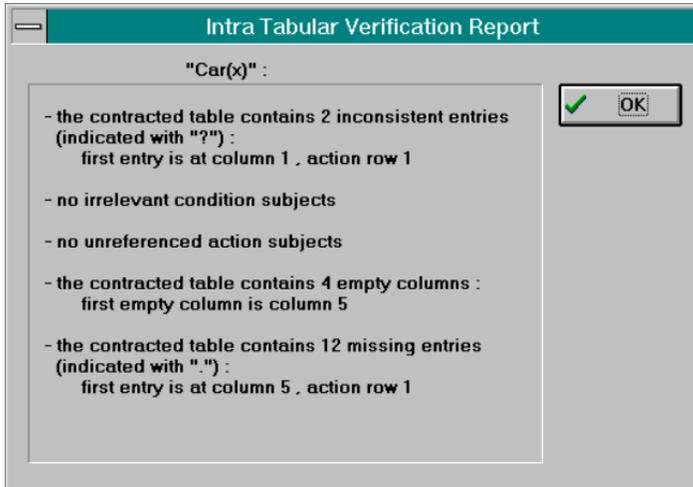


Figura 2.12: Reporte de verificación intra-tabular Car(x)

Análisis de la tabla look(x)

Es una subtabla de condiciones para determinar el valor de la condición look(x) de la tabla Car(x) que se analizó anteriormente, los valores que puede tomar son “True” o “False”. La figura 2.13 muestra el editor de PROLOGA para la construcción de la tabla look(x).

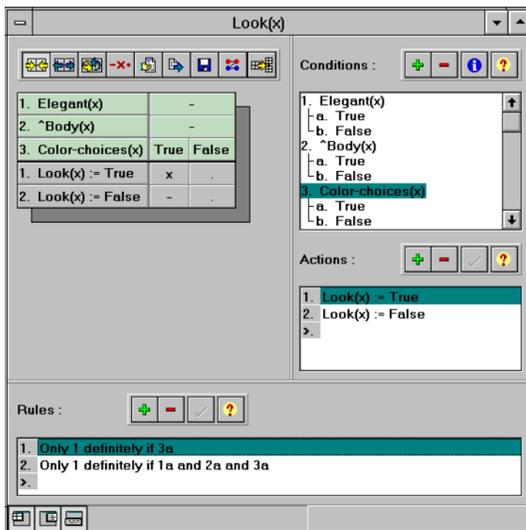


Figura 2.13: Editor de la tabla look(x)

Las condiciones y las acciones son indicadas por números, los estados de las condiciones por letras. Las entradas de las acciones pueden ser llenadas directamente dando clic con el mouse sobre la tabla o se puede optar por escribir las reglas de entrada en la lista de la parte inferior de la ventana. En la figura 2.13 se muestra la tabla de decisión correspondiente a las reglas C24 y C30 del ejemplo que se ha venido analizando.

- C24: look(x) ←--- color_choices(x)
- C30: look(x) ←--- elegant(x), body_chape(x), color_choices(x)

La regla C24 subsume la regla C30, es decir, abarca mayor cantidad de casos en su interior ya que solamente requiere de que una condición tenga un valor determinado que es la de color_choice(x)=True. Internamente PROLOGA trata este tipo de anomalía dentro de un tipo de tabla que le llama “single-hit table”, en la cual todas las columnas son mutuamente excluyentes, lo que provoca que la anomalía no persista en la tabla de decisión final. PROLOGA garantiza la exclusividad en las columnas de forma automática construyendo un árbol estructurado de tablas, basado en una combinación ordenada de estados de condiciones relevantes procediendo de tal manera que los estados de menos condiciones varían primero (ver figura 2.14).

single-hit table :

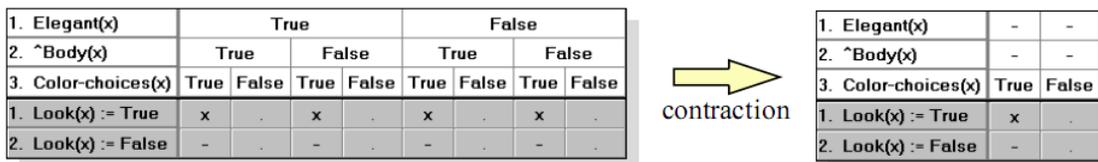


Figura 2.14: Ejemplo de subsumir en una tabla single-hit.

Ahora analicemos en reporte de verificación intra-tabular de la tabla look(x) (ver figura 2.15) donde se confirma de que las condiciones elegant(x) y body(x) son irrelevantes ya que no influyen en la realización de la acción. Una condición se vuelve irrelevante si su estado no influye en que las acciones puedan llevarse a cabo. Cuando esto ocurre la fila de toda condición irrelevante se vuelve redundante por lo tanto hay que eliminarla (Vanthienen et al., 2009).

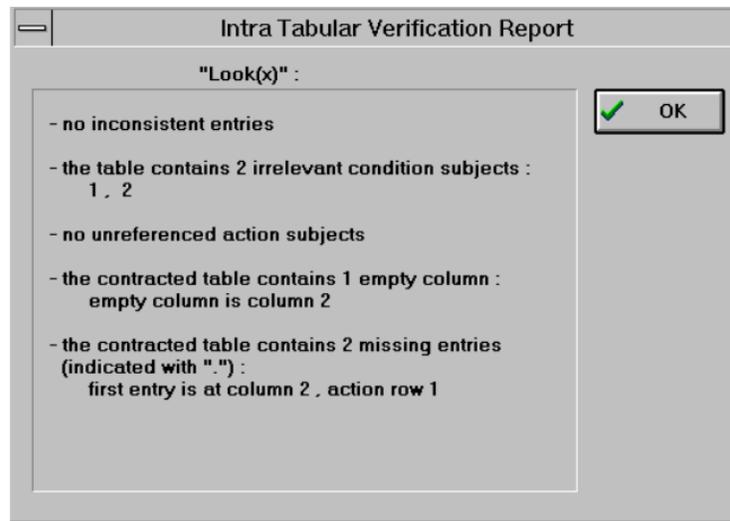


Figura 2.15: Reporte de verificación intra-tabular de la tabla look(x).

Existen otros tipos de anomalías que se pueden detectar a través de las tablas de decisión, en la siguiente sesión se explicarán en qué consisten cada una de ellas.

2.3.4 Otros tipos de anomalías

En esta sesión se explicarán otros tipos de anomalías detectadas con la herramienta PROLOGA (Vanthienen et al., 2009, Vanthienen and Wets, 1995).

Redundancia

- Columnas duplicadas: Ocurre cuando un par de reglas coinciden en valores en las condiciones y en las acciones. En la definición de tabla de decisión que utiliza PROLOGA se utiliza el criterio de exclusividad entre las columnas de una misma tabla de decisión, lo cual previene que ocurra este tipo de redundancia.
- Columnas subsumidas: Una columna subsume a otra cuando engloba un conjunto de casos más amplio que el de la otra, es decir, que requiere del chequeo de menos condiciones, por lo que tiene incluida a la otra. El criterio que utiliza PROLOGA de que las columnas sean mutuamente excluyentes hace de que no puedan ser introducidas reglas que generen este tipo de redundancia.
- Fila de condición redundante: Se presenta cuando una determinada fila no influye en la determinación de la acción al comparar dos o más reglas, por lo

tanto esta fila es irrelevante. Se manifiesta también cuando es redundante con otra fila de condición.

- Fila de acción redundante: Se detecta cuando entra en redundancia con otra fila de acción, lo cual puede prevenirse durante el proceso de construcción de la tabla de decisión.

Ambivalencia (Vanthienen and Wets, 1995)

- Ambivalencia a nivel de columna: Ocurre cuando hay columnas en una tabla de decisión que son totalmente aplicables a un determinado subconjunto de datos de entradas, sin embargo resultan tener conclusiones contradictorias bajo las mismas condiciones. En PROLOGA no ocurre si se emplea correctamente el criterio de exclusividad, se emite un error inmediatamente que son introducidas las reglas que se encuentran en conclusiones contradictorias.
- Ambivalencia a nivel de fila: Ocurre cuando dos o más filas tienen entradas contradictorias en la misma columna.
- Reglas ambiguas: Reglas con iguales valores de condiciones y con diferentes, pero no contradictorias, acciones, no se sabe exactamente cuál acción realizar.

Circularidad

Es un tipo de anomalía inter-tabular que ocurre cuando se presentan cadenas circulares de tablas de decisión dentro de un mismo proyecto. Son un conjunto de reglas donde se produce una conclusión en alguna parte como una de las premisas. En PROLOGA se detectan mediante la comprobación de ciclos en el grafo estructurado inter-tabular (Vanthienen and Wets, 1995).

Deficiencia (Vanthienen et al., 2009, Vanthienen and Wets, 1995)

- Incompletitud: Es causado por reglas aisladas que corresponden a tablas no usadas.
- Acciones no referenciadas: Se evidencia cuando existen estados que nunca pueden ser alcanzados, sin embargo, son estados válidos.

- Falta de reglas: Se presentan cuando hay varias columnas vacías, lo que indica que hay valores válidos de las condiciones que no son referenciados en las tablas de decisión. Con el criterio de completitud que emplea PROLOGA asegura de que en el nivel intra-tabular, todos los posibles estados se incluyen en la parte de las condiciones, colocando la debida acción correspondiente en la parte de las acciones.
- Valores de atributos no utilizados o combinaciones de estos: Se evidencia su presencia cuando los valores de determinados atributos (o combinaciones de estos) nunca ocurren como condiciones, es decir, las reglas asociadas nunca se pueden ejecutar, y es debido a que faltan una serie de reglas que tengan esas condiciones reformuladas. La naturaleza de las tablas de decisión permite fácilmente verificar la integridad de las reglas debido a que el número de columnas simples en la matriz de valores de las condiciones debe ser igual al producto del número de estados de cada condición. La instrucción de combinación garantiza generar todas las posibles combinaciones de los estados de las condiciones. Esta garantía de la integridad de las combinaciones de estado es una de las principales ventajas de las tablas de decisión, PROLOGA lo tiene en cuenta por eso es que no ocurre este tipo de deficiencia en su interior (Vanthienen et al., 2009, Vanthienen and Wets, 1995).

Seguidamente se realizará un análisis de los puntos comunes y diferentes entre los dos métodos de validación de la consistencia de RN basadas en tablas de decisión: el propuesto por Josefina y el de PROLOGA de Jan Vanthienen.

2.4 Análisis de ambas vías para validación de la consistencia usando tablas de decisión

Después de expuestos los aspectos de cada uno de los métodos propuestos, uno propuesto por María Josefina Castilla (Castilla, 2007) y el otro de Jan Vanthienen (Vanthienen, 2003, Vanthienen et al., 2009) plasmado en el software PROLOGA se quiere realizar una exposición acerca de las similitudes y diferencias que existen entre los dos métodos según diferentes criterios.

Criterios de comparación

- Definición de las tablas de decisión.
- Estructura de las tablas de decisión.
- Tipos de reglas que se pueden representar en las tablas de decisión.
- Completitud o integridad de las tablas de decisión.
- Tipos de tablas de decisión.
- Tipos de anomalías que detecta.
- Formas de eliminar las redundancias.
- Lugar donde se presentan las anomalías que detecta.

Similitudes

- Definen de igual forma las tablas de decisión como un instrumento u objeto que sintetiza procesos para la toma de decisiones.
- Estructura de las tablas de decisión, es una tabla con cuatro cuadrantes: identificación de condiciones, entrada de condiciones, identificación de acciones y entrada de acciones.
- Tipos de reglas: puras y mixtas.
- Completitud o integridad de las tablas de decisión.
- Tipos de tablas de decisión según los valores de las condiciones: estradas limitadas, entradas ampliadas y mixtas.
- Detectan anomalías de ambivalencia a nivel de columna, a nivel de fila y reglas ambiguas; redundancia por columna duplicada, fila de condición redundante y fila de acción redundante; deficiencia por acciones no referenciadas.

Diferencias

- La anomalía de redundancia por columnas subsumidas es tratada de forma diferente por ambos métodos ya que, en el método de Josefina se elimina a

través de la segunda ley que clasifica las reglas y las que sean dependientes y redundantes donde una es pura y la otra es mixta, la pura está contenida en la mixta por lo que es eliminada de la tabla. Por otra parte el sistema PROLOGA no admite la entrada de reglas que se subsumen por el criterio de exclusividad cuando se está construyendo la tabla de decisión.

- Trata nada más las anomalías e inconsistencias que ocurren dentro de la misma tabla de decisión, es decir, intra-tabular, sin embargo PROLOGA trata anomalías intra-tabular e inter-tabular.
- PROLOGA detecta otras anomalías que no las detecta la otra vía, como son circularidad, deficiencias por incompletitud, falta de reglas, valores de atributos no utilizados o combinaciones de estos y acciones no referenciadas; debido a que son anomalías del tipo inter-tabular.

Criterios de comparación	Método de Josefina	Método de Jan (PROLOGA)
Definición de las tablas de decisión	Definición similar	Definición similar
Estructura de las tablas de decisión	Igual	Igual
Tipos de reglas que se pueden representar en las tablas de decisión	Los mismos tipos de reglas	Los mismos tipos de reglas
Compleitud o integridad de las tablas de decisión	Es tratada	Es tratada
Tipos de tablas de decisión	Entradas reducidas, ampliadas y mixtas	Entradas reducidas, ampliadas y mixtas

Tipos de anomalías que detecta	Ambivalencia, redundancia y deficiencias por acciones no referenciadas.	Ambivalencia, redundancia, deficiencias por acciones no referenciadas, deficiencias por incompletitud, falta de reglas, valores de atributos no utilizados o combinaciones de estos y acciones no referenciadas.
Formas de eliminar las redundancias	Elimina la redundancia por columna subsumida a través de la segunda ley.	No admite la entrada de reglas que tengan redundancia por columna subsumida por el criterio de exclusividad donde cada regla es única.
Lugar donde se presentan las anomalías que detecta	Trata anomalías que ocurren dentro de una misma tabla (intra-tabular).	Trata anomalías que ocurren dentro de una misma tabla (intra-tabular) y entre tablas diferentes (inter-tabular).

2.5 Algoritmo para la validación de la consistencia del conjunto de reglas de negocios

Después de aplicado el método de validación de la consistencia de RN basado en tablas de decisión, y visto su efecto, en que diferentes tipos de anomalías desde el punto de vista de la consistencia que se pueden presentar en las RN, se pasa a la descripción de cada uno de sus pasos.

2.5.1 Estructura del documento de entrada

El algoritmo recibe como entrada un documento *.txt con las reglas definidas con la estructura siguiente:

If condición then acción

donde <condición> es una condición o conjunto de condiciones separadas por coma (,) en forma de una sola palabra; <acción> es una acción o conjunto de acciones separadas por coma (,) en forma de una sola palabra; por ejemplo:

if OtrasEnferm,CondFisFav,\$SiClasif

then AnalisisEnfermAfectanTR,InsertaListaEspera

Si una condición no se referencia en una regla significa que el valor que toma es “N” y si se referencia antecedida con el símbolo “\$” significa que no importa el valor que tome dicha condición, en la tabla de decisión se representa con una indiferencia (-). Por ejemplo, la regla anterior se representaría en la tabla de decisión como sigue:

¿Presenta otras enfermedades?	S
¿Condiciones Físicas Favorables?	S
¿Se conoce clasificación del Paciente?	-
Análisis de enfermedades que afectan el TR	x
Realizar análisis complementarios	-

Inserción en lista de espera	x
Remitir consulta de progresión	-

2.5.2 Descripción general

El algoritmo parte de la captura de las diferentes condiciones y acciones en cada RN, normalizando el lenguaje con el objetivo de analizar y realizar una depuración identificando contradicciones (cuya solución está en la revisión de la captura de las RN) y redundancias. Tales redundancias pueden ser eliminadas con la aplicación de tres leyes y un criterio expuestos en el epígrafe 1.4.3.5.

Paso 1: Editar las reglas en un documento *.txt a partir del enunciado.

Paso 2: Captar el conjunto de reglas a partir del documento *.txt y construir la tabla de decisión.

Paso 3: Verificar si existen ambivalencias o contradicciones entre las reglas.

Paso 4: Aplicar primera ley: unificación de reglas por indiferencia.

Paso 5: Aplicar criterio de clasificación de pares de reglas.

Paso 6: Aplicar segunda o tercera ley de depuración.

Paso 7: Salida de reglas validadas en un fichero *.txt.

2.5.3 Descripción detallada

Seguidamente se detallan cada uno de los pasos del algoritmo enunciados anteriormente.

Paso 1: Se identifican las condiciones y acciones a partir del enunciado y se construyen en un documento *.txt con el formato definido en el epígrafe 2.5.1.

Paso 2: Cargar el conjunto de reglas editadas en el documento *.txt para ser analizadas y construir la tabla de decisión. Las reglas que contengan el signo de peso (\$) delante de las condiciones son consideradas como indiferentes se les debe poner el símbolo

indiferencia (-) y para las condiciones que no aparezcan son tomadas como negativas (N) en la tabla de decisión.

Paso 3: Realizar el proceso de verificar si existen ambivalencias o contradicciones. En el mismo se analizan las reglas que tengan iguales entradas de condiciones y en caso que contengan diferentes entradas en las acciones se imprime que se encontró una contradicción entre tales reglas (1, 2, ...) y termina el algoritmo pues estas no se pueden eliminar.

Paso 4: Aplicar la primera ley: Unificación de reglas por indiferencia, donde se analizan todos los pares de reglas posibles, y se identifican aquellos pares que tengan idénticas entradas en las condiciones y acciones excepto en una condición donde una registra una entrada S (sí) y la otra registra una entrada N (no). Los pares de reglas que cumplen esta condición, pueden ser eliminadas y sustituidas por una sola regla con idénticas entradas de condiciones y acciones que registre la entrada indiferente (\$) en aquella condición donde las originales diferían.

Paso 5: Todos los pares de reglas resultantes del paso 4 se clasifican en dependientes o independientes basado en un criterio de clasificación. Dos reglas son independientes si en las entradas de las condiciones no existe cambio del valor “S” por “N” o viceversa, si no cumple con este criterio son dependientes. Los pares de reglas dependientes pueden ser redundantes o contradictorias; las dependientes redundantes son las que tienen iguales acciones y las contradictorias son las que tienen acciones diferentes que fueron eliminadas en el paso 3.

Paso 6: En el conjunto de reglas resultante del paso 5 se buscan los pares de reglas dependientes y redundantes, si una regla es pura y la otra es mixta, se aplica la segunda ley para eliminar la pura ya que está contenida en la mixta; en caso de que ambas reglas sean mixtas, se aplica la tercera ley y se busca la regla pura común a ambas y se elimina de una de ellas, quedando como resultado una regla pura y otra mixta que no contienen la regla pura común.

Paso 7: Salida de reglas validadas en un fichero con el nombre Validada_*.txt que muestra las reglas resultantes sin inconsistencias.

En el anexo 2 se muestra un pseudocódigo del algoritmo descrito.

2.6 Tipos de reglas que se puede aplicar el algoritmo

En el enfoque de RN las reglas que se clasifican como reglas de producción, son las utilizadas para ser validadas con el algoritmo implementado usando las tablas de decisión. Estas reglas, denominadas producciones, son una representación básica encontrada útil en la planificación automatizada, sistemas expertos y selección de acciones.

Un sistema de producción proporciona el mecanismo necesario para ejecutar producciones a fin de lograr una meta para el sistema. Las reglas que representan el conocimiento con estos sistemas utilizan un formato *SI-ENTONCES (IF-THEN)*, es decir tienen 2 partes:

- La parte *SI (IF)*, es el antecedente, premisa, condición o situación.
- La parte *ENTONCES (THEN)*, es el consecuente, conclusión, acción o respuesta.

Considere el ejemplo siguiente:

R1: 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.

Esta regla determina si se incluirá al paciente en la lista de espera para los trasplantes su tratamiento específico mediante la aplicación de lógica de negocios, en forma de una comparación utilizando la edad del paciente.

Se construye la tabla de decisión en la cual se va a realizar el análisis de las reglas, agrupando estas por las condiciones y acciones que realizan, las reglas que chequeen las mismas condiciones y realicen determinadas acciones, que tengan cierta relación. Esa relación entre las acciones que realizan pudiera ser que sean contrarias, o que realicen más de una con cierta precedencia entre las mismas, para facilitar la detección de anomalías que con las reglas de forma aislada no son fáciles de detectar.

2.7 Conclusiones parciales del capítulo

En este capítulo se ha realizado un análisis de dos formas de validación de la consistencia de un conjunto de RN basadas en tablas de decisión, de lo cual se ha llegado a las siguientes conclusiones:

- 1 Se describe formas de manejo de las tablas de decisión para realizar la validación de la consistencia de un conjunto de reglas de negocio.
- 2 Se realiza una valoración comparativa de dos formas de trabajo con las tablas de decisión basadas en diferentes criterios, identificando similitudes y diferencias.

Criterios de comparación

- Definición de las tablas de decisión.
- Estructura de las tablas de decisión.
- Tipos de reglas que se pueden representar en las tablas de decisión.
- Completitud o integridad de las tablas de decisión.
- Tipos de tablas de decisión.
- Tipos de anomalías que detecta.

Similitudes

- Definen de igual forma las tablas de decisión como una herramienta que sintetiza procesos para la toma de decisiones.
- Estructura de las tablas de decisión.
- Tipos de reglas: puras y mixtas.
- Completitud o integridad de las tablas de decisión.
- Tipos de tablas de decisión según los valores de las condiciones: estradas limitadas, entradas ampliadas y mixtas.
- Detectan anomalías de ambivalencia a nivel de columna, a nivel de fila y reglas ambiguas; redundancia por columna duplicada, fila de condición

redundante y fila de acción redundante; deficiencia por acciones no referenciadas.

Diferencias

- La anomalía de redundancia por columnas subsumidas es tratada de forma diferente por ambos métodos ya que, en el método de Josefina se elimina a través de la segunda ley que clasifica las reglas y las que sean dependientes y redundantes donde una es pura y la otra es mixta, la pura está contenida en la mixta por lo que es eliminada de la tabla. Por otra parte el sistema PROLOGA no admite la entrada de reglas que se subsumen por el criterio de exclusividad cuando se está construyendo la tabla de decisión.
 - Trata nada más las anomalías e inconsistencias que ocurren dentro de la misma tabla de decisión, es decir, intra-tabular, sin embargo PROLOGA trata anomalías intra-tabular e inter-tabular.
 - PROLOGA detecta otras anomalías, como son circularidad, deficiencias por incompletitud, falta de reglas, valores de atributos no utilizados o combinaciones de estos y acciones no referenciadas; debido a que son anomalías del tipo inter-tabular.
- 3 Se obtiene una vía para la validación de la consistencia de un conjunto de reglas de negocio, mediante un algoritmo del cual se describen los pasos en un lenguaje en pseudocódigo.

Capítulo III: Validación de reglas de negocio para el caso de estudio

Realizar la validación de RN en el repositorio de reglas existentes es el propósito fundamental de esta investigación. En el presente capítulo se realiza una caracterización del caso de estudio para el área de nefrología desde el *enfoque de RN*. Se muestra un ejemplo de cómo llevar a cabo el proceso de validación de RN dentro del repositorio para que este se mantenga consistente, sin la presencia de anomalías. Se brinda una descripción detallada de los pasos a seguir para realizar la validación, es decir, el algoritmo de trabajo implementado en una aplicación de software.

3.1 Caso de estudio: área de nefrología

La necesidad de obtener sistemas informáticos capaces de ayudar a los trabajadores de salud en los procedimientos durante el seguimiento a pacientes de nefrología hace útil la aplicación de RN en los sistemas de información. Los pacientes son tratados en esta área de la medicina donde la práctica del trasplante ha sido adoptada en los últimos años como primera opción terapéutica. Cada procedimiento a seguir durante el tratamiento es regido por complejos protocolos de trabajo. El paciente puede pasar por diferentes tipos de consultas según sean sus avances y evolución particular. Médicos y paramédicos deben tomar decisiones importantes ante cada nueva situación según los resultados de los exámenes realizados a sus pacientes y de los protocolos que rigen todo el proceso.

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 telecomunicaciones a la salud, tiene como objetivo principal prestar servicio a los profesionales de la salud para mejorar la calidad en la atención médica; representa la intersección de las ciencias de la información, ciencias de la computación y la atención de la salud. Esta rama de la informática 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 solo los ordenadores, sino también guía de práctica clínica, terminología médica formal y, de sistemas de información (SI) y comunicación. En ocasiones es considerada como una especialidad médica, debido a que es el estudio del conocimiento médico y su aplicación en el manejo del paciente a través de SI y telecomunicaciones que requiere de una extensa base de formación médica, clínica y de las posibilidades de la tecnología de distribuir esta información en forma rápida y efectiva

3.1.1 Descripción

La insuficiencia total o casi total en el funcionamiento del riñón para excretar los desechos, concentrar la orina y regular la electrólisis es considerada como *insuficiencia renal crónica (IRC)*. Cuando esta enfermedad está en una etapa terminal, que se considera el desenlace común a múltiples enfermedades que afectan el riñón, se denomina *insuficiencia renal crónica terminal (IRCT)*. Esta se presenta cuando los riñones ya no pueden funcionar al nivel necesario para la vida diaria, es decir, que la IRC progresa a un punto tal que la función de los riñones es menor del 10% de su capacidad normal. El área de nefrología atiende a los pacientes de forma especializada, insertándose en un sistema de atención secundaria y terciaria personalizada. Se requieren registros permanentes de los pacientes que faciliten gestionar los exámenes y la evolución para cada enfermo. Dicha área incluye diversas especialidades que, en dependencia de cada uno de estos controles, tiene sus peculiaridades. La atención al paciente se realiza de acuerdo al seguimiento en tres tipos de consultas (Martínez Busto et al., 2010b):

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

Cada consulta tiene sus propios requerimientos 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.

La insuficiencia total o casi total en el funcionamiento del riñón para excretar los desechos, concentrar la orina y regular la electrólisis es considerada como *insuficiencia renal crónica (IRC)*. Cuando esta enfermedad se encuentra en una etapa donde el riñón está en muy mal estado respecto a su funcionamiento, es decir, en una fase terminal, que se considera el desenlace

común a múltiples enfermedades que afectan al riñón, se denomina *insuficiencia renal crónica terminal* (IRCT). Esta se presenta cuando los riñones ya no pueden funcionar al nivel necesario para llevar una vida normal, es decir, que la IRC avanza a un punto tal que la función de los riñones es menor del 10% de su capacidad normal. El área de nefrología atiende pacientes en los diferentes *estadios* de esta enfermedad.

Todo paciente con diagnóstico de IRCT, de preferencia en estadio predialítico o una vez que esté siendo atendido con un método depurador, debe ser valorado en una *consulta de nefrología* en su área de atención. Allí se explica al paciente las opciones de tratamiento, incluido el TR.

Los posibles *estadios* de un paciente se encuentran entre los valores del I al V. De encontrarse el paciente en algunos 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 del paciente es V se considera 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*.

El recibir terapia con *diálisis peritoneal* puede representar una opción para salvar la vida de aquellos pacientes con IRCT que no pueden, por alguna razón, recibir tratamiento de trasplante del órgano que puede provenir de un donante familiar vivo, donante muerto o donante con muerte encefálica (Kasiske et al., 1995, Ojo et al., 2001).

Al decidir el origen del órgano se establecen dos tipos posibles de donantes para conformar la pareja *receptor-donante potencial*⁷, considerando que es posible tener uno o varios donantes potenciales y uno o varios posibles receptores. La pareja constituida es sometido a un *protocolo de compatibilidad*.

En caso de tener un resultado favorable, al finalizar el *protocolo de compatibilidad* se envían los pacientes a la *consulta de TR* o *consulta multidisciplinaria*, o sea al receptor y posibles donadores o donantes potencial donde evalúan la pareja *receptor-donador potencial*.

3.1.2 Caracterización del caso de estudio

De todo lo anteriormente descrito sobre el proceso de llevar a cabo el TR se concluye que:

⁷ Se considera “receptor-donante potencial” o “receptor-donador potencial” a la pareja que se integra por un receptor con un posible donante de riñón.

1. Durante la atención a un paciente en esta área se tienen que tomar múltiples decisiones que están muy bien reglamentadas en diferentes protocolos.
2. Las decisiones están fuertemente interrelacionadas, o sea, unas dependen de otras.
3. Cada usuario, médico, paramédico, paciente o familiar tiene requerimientos informativos diferentes. Debe propiciarse el acceso y manejo de la información a cada cual según su rol.
4. Los tratamientos y procedimientos son susceptibles a cambios, e incluso hasta pueden variar de acuerdo a la región o país. Se rigen por normativas o avances científicos a diferentes niveles.
5. Se debe permitir a usuarios avanzados a las políticas trazadas en un ambiente flexible, este tipo de usuario puede ser incluso, el propio médico.

Lo anterior sugiere que la aplicación del *enfoque de RN* facilita la obtención de soluciones computacionales para este dominio. Debido, fundamentalmente, a la elevada probabilidad de cambios rápidos y constantes que afectan el ambiente de la aplicación, identificándose la necesidad de renovación y adaptación.

El sector de la salud es un área en que los cambios son muy frecuentes dado el vertiginoso desarrollo que tiene la medicina, tanto en nuestro país como a nivel mundial. Estos cambios se originan en la comunidad científica y se canalizan a través del sistema de salud hasta su instrumentación por parte de los especialistas, que incluso pueden aportar, con su experiencia, determinadas variaciones. Esto refuerza que deben ser los propios especialistas del negocio, en este caso los médicos, los encargados de crear y dar mantenimiento a sus políticas, expresadas con una terminología que depende en gran medida de cada área o sector. Hay que tener en cuenta las reglamentaciones y normativas que rigen o restringen los diferentes procesos y actividades en un negocio, no solo en la etapa inicial del ciclo de vida de un SI, sino también a lo largo del resto de las diferentes etapas, especialmente en sectores tan sensibles como el de la salud.

3.2 Captura del conjunto de reglas usando UCRules

Las herramientas computacionales brindan muchas facilidades para la administración de RN. Mediante ellas se logra la separación de las reglas tanto de la lógica de los datos como de las funcionalidades de la aplicación, facilitando el acceso, modificación y administración de las RN, tanto a desarrolladores como a usuarios del negocio.

En la investigación (Martínez Busto, 2011) se propuso una herramienta independiente de la base de datos llamada “UCRules” con la función de administrar las RN.

3.2.1 Herramienta UCRules

La captación de las RN para este caso de estudio se realiza a través de una herramienta confeccionada por el grupo de trabajo de Base de Datos llamada “UCRules” (ver figura 3.1). Está integrada por varios módulos, los que realizan el manejo de las reglas desde la visión del propietario del negocio, facilitando el trabajo de los diseñadores e implementadores (Martínez Busto et al., 2010b).

El editor de “UCRules” permite a los expertos del negocio manejar y construir las reglas a través de una interfaz gráfica según los patrones definidos en el epígrafe 2.1. Previamente el analista del negocio deberá definir el vocabulario del negocio sobre el modelo de hechos genérico con enfoque ontológico que brinda la herramienta. El panel izquierdo superior muestra un árbol jerárquico con diferentes categorías según la clasificación semántica; debajo se encuentran las reglas en proceso de edición y en el panel derecho aparece una pestaña para cada una de ellas. En dichas pestañas se introducen las reglas en sí, siendo la esencia de la actividad de captura de RN.

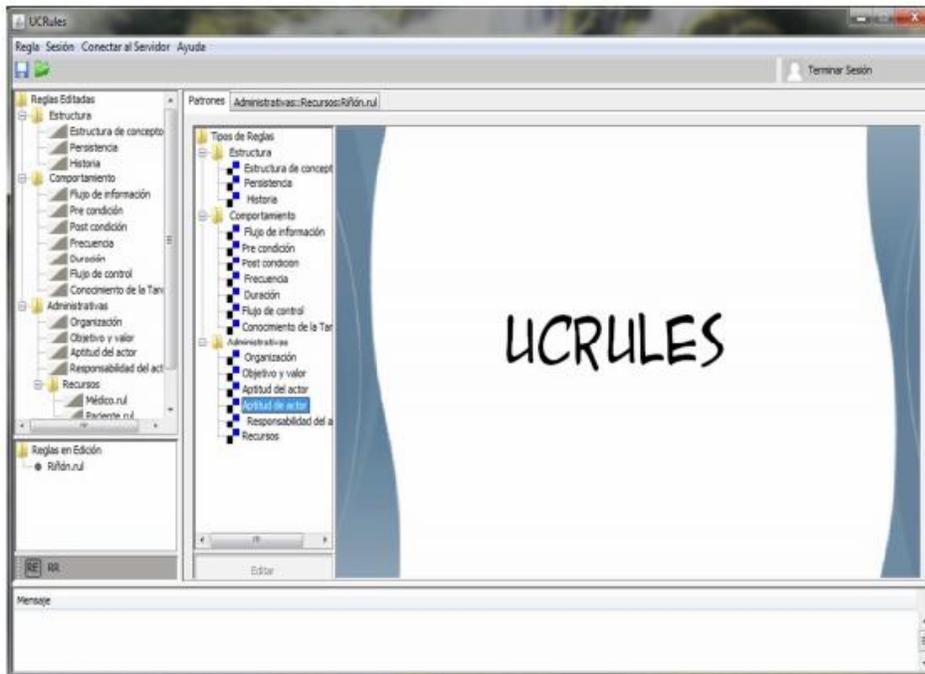


Figura 3.1: Interfaz principal del editor de “UCRules”.

Durante la captura de las reglas se puede acceder a los diferentes patrones de RN haciendo uso de la pestaña especial del panel derecho, ubicada a la izquierda y llamada Patrones. Esto tiene como objetivo permitir, al que está realizando la edición, valorar la conveniencia del uso de un determinado patrón.

3.2.2 Conjunto de reglas de negocio

El conjunto de RN puede verse como un punto en común entre la comunidad de usuarios del negocio y los profesionales de cómputo, véase la figura 3.2. Los primeros introducen las reglas y, una vez validadas, pasan al repositorio que permite a los desarrolladores interpretarlas e implementarlas. Esta entrada y validación se realiza mediante el editor de RN, basado en los patrones de diseño de las reglas (Boggiano Castillo et al., 2007a, Boggiano Castillo et al., 2007b, Boggiano Castillo et al., 2009a).

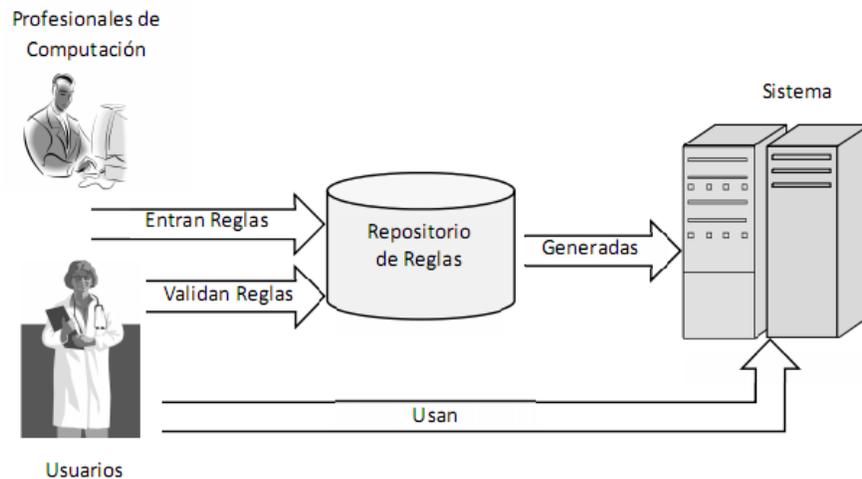


Figura 3.2: Diagrama de intercepción entre usuarios y profesionales de computación (Boggiano Castillo et al., 2010)

Una vez introducidas las RN al repositorio que cumplan con determinado patrón de diseño, es decir, validadas estructuralmente, se generan automáticamente mediante módulos en los que trabaja el grupo de investigación de BD. Las RN después de generadas, dentro del repositorio pueden presentar diferentes tipos de anomalías como las explicadas en el epígrafe 2.3.1. La validación del repositorio de RN para eliminar dichas anomalías es el objetivo fundamental de la presente investigación. El método para realizar dicha validación está basado en tablas de decisión como se explicará, con un ejemplo práctico, en el siguiente epígrafe.

3.3 Aplicación del algoritmo de validación para el caso de estudio

El método de validación de la consistencia considerado durante la presente investigación, permite realizar el análisis lógico de las RN almacenadas en un conjunto, está basado en tablas de decisión. A continuación se explica la aplicación del algoritmo a través de un ejemplo práctico sobre el caso de estudio descrito en el epígrafe 3.1.

3.3.1 Conjunto de reglas de negocio

Basado en el conjunto de RN para poner en práctica el algoritmo de trabajo y realizar la validación de la consistencia del mismo, se escogieron determinadas 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.

Las reglas son las siguientes:

- R1:** 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.
- R2:** 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.
- R3:** 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.
- R4:** Si el paciente presenta condiciones físicas favorables, no padece de otras enfermedades y 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.
- R5:** Si del paciente no se conoce su clasificación se remite a consulta de progresión.
- R6:** Si el paciente presenta condiciones físicas favorables y no padece de otras enfermedades se le realiza análisis complementarios.
- R7:** Si el paciente no presenta condiciones físicas favorables y no se conoce su clasificación se le realizan análisis complementarios.

R8: Si el paciente no padece de otras enfermedades y no se conoce su clasificación se le realizan análisis complementarios.

3.3.2 Construcción de la tabla de decisión

Ahora se construye la tabla de decisión, capturando las diferentes condiciones y acciones en cada regla, normalizando el lenguaje con el objetivo de analizar y realizar una depuración de las reglas identificando contradicciones y redundancias. Una vez puestas cada una de las condiciones se construyen todas las posibles combinaciones de valores de las condiciones que se puedan presentar y se coloca en la parte superior derecha. La cantidad de columnas se halla multiplicando la cantidad de posibles valores de cada una de las condiciones, en este caso cada condición solo puede tomar dos valores, S o N, por lo que este producto es, $2 \times 2 \times 2 \times 2 = 2^4 = 16$.

Tabla de decisión: condiciones y acciones

¿Presenta otras enfermedades?	S	S	N	N	S	N	-	N
¿Condiciones Físicas Favorables?	S	N	N	-	S	S	N	-
¿Se conoce clasificación del Paciente?	S	S	S	S	N	N	N	N
Análisis de enfermedades que afectan el TR	x	-	-	-	x	-	-	-
Realizar análisis complementarios	-	x	x	x	-	x	x	x
Inserción en lista de espera	x	x	x	x	-	-	-	-
Remitir consulta de progresión	-	-	-	-	x	x	x	x
	1	2	3	4	5	6	7	8

3.3.3 Depuración de la tabla de decisión

Mediante este proceso de logra:

1. Hacer más compacta la tabla, es decir, que tenga menos reglas.
2. Encontrar posibles errores en las reglas.

Se realiza a través de tres leyes y un criterio, que se enunciaron en el epígrafe 1.4.1.4.

Primera ley: Unificación de reglas por indiferencia (Castilla, 2007).

Se analizan todos los pares de reglas posibles, con el objetivo de buscar reglas que tengan iguales entradas de condiciones y acciones pero que solamente difieran en una condición donde una registre una entrada “N” y la otra una entrada “S”. Las que cumplan esta condición se eliminan de la tabla y se remplazan por una regla que registre una indiferencia (-) en la entrada de condiciones donde diferían las reglas que realizaban las mismas acciones. El tipo de anomalía que se elimina con este paso es *fila de condición redundante o irrelevante* ya que su valor no influye en la acción que se va a realizar.

Aplicando esta ley al ejemplo se halla que las reglas 2 y 3 cumplen con esta condición, la tabla queda como sigue:

¿Presenta otras enfermedades?	S	-	N	S	N	-	N
¿Condiciones Físicas Favorables?	S	N	-	S	S	N	-
¿Se conoce clasificación del Paciente?	S	S	S	N	N	N	N
Análisis de enfermedades que afectan el TR	x	-	-	x	-	-	-
Realizar análisis complementarios	-	x	x	-	x	x	x
Inserción en lista de espera	x	x	x	-	-	-	-
Remitir consulta de progresión	-	-	-	x	x	x	x
	1'	2'	3'	4'	5'	6'	7'

Una vez aplicada la primera ley se aplica un criterio.

Criterio: Reglas independientes y dependientes

Con la tabla resultante se forman todos los pares de reglas posibles para clasificar esos pares de reglas en independientes o dependientes, atendiendo a que si por lo menos para una condición

una regla registra una entrada “S” y la otra registra una entrada “N”, se dice que ese par de reglas son *independientes*, es decir que no presentan ninguna anomalía; si no cumple con ese criterio se dice que son *dependientes*. Si el par de reglas son *dependientes* es posible determinar si son *redundantes* o *contradictorias*. Un par de reglas dependientes son *redundantes* cuando describen las mismas acciones, es decir, cuando presentan las anomalías de redundancia con columnas duplicadas o par de columnas subsumidas; y *contradictorias* cuando describen diferentes acciones, con la anomalía de ambivalencia tanto de columna como de fila.

En el ejemplo se van a clasificar los pares de reglas en dependientes o independientes.

1' y 2': son independientes	3' y 4': son independientes
1' y 3': son independientes	3' y 5': son independientes
1' y 4': son independientes	3' y 6': son independientes
1' y 5': son independientes	3' y 7': son independientes
1' y 6': son independientes	4' y 5': son independientes
1' y 7': son independientes	4' y 6': son independientes
2' y 3': son dependientes redundantes	4' y 7': son independientes
2' y 4': son independientes	5' y 6': son independientes
2' y 5': son independientes	5' y 7': son dependientes redundantes
2' y 6': son independientes	6' y 7': son dependientes redundantes
2' y 7': son independientes	

Si los pares de reglas son dependientes redundantes puede aplicarse la segunda y tercera ley, sino significa que hay un error en el enunciado o se ha construido mal la tabla, es decir, que presenta anomalía de ambivalencia tanto de fila como de columna.

Eliminación de redundancias

Segunda ley

Si 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. Se elimina la anomalía de redundancia de par de columnas subsumidas ya que la regla mixta subsume a la regla pura.

Los siguientes pares de reglas son dependientes y redundantes:

2' y 3': ambas son mixtas, por lo tanto se le aplica la *tercera ley*

6' y 7': ambas son mixtas, por lo que se le aplica la *tercera ley*

5' y 7': la 5' es pura y la 7' es mixta, por lo que se le aplica la *segunda ley*

Se colocan las dos reglas de forma continua descomponiendo la mixta en sus respectivas reglas puras correspondientes:

5'	7'a	7'b
N	N	N
S	S	N
N	N	N

La regla 5' está contenida en la 7'a, por lo tanto puede ser eliminada de la tabla quedando como sigue:

¿Presenta otras enfermedades?	S	-	N	S	-	N
¿Condiciones Físicas Favorables?	S	N	-	S	N	-
¿Se conoce clasificación del Paciente?	S	S	S	N	N	N
Análisis de enfermedades que afectan el TR	x	-	-	x	-	-
Realizar análisis complementarios	-	x	x	-	x	x
Inserción en lista de espera	x	x	x	-	-	-
Remitir consulta de progresión	-	-	-	x	x	x
	1''	2''	3''	4''	5''	6''

Tercera ley

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.

Las reglas 2'' y 3'' son dependientes redundantes y mixtas, y las reglas 5'' y 6'' son también dependientes redundantes y mixtas, por lo que se les aplica la *tercera ley* para eliminar la redundancia. La anomalía presentada en estos casos es redundancia con columnas duplicadas.

Descomponemos las reglas mixtas en las puras correspondientes:

2''a	2''b	3''a	3''b	5''a	5''b	6''a	6''b
S	N	N	N	S	N	N	N
N	N	S	N	N	N	S	N
S	S	S	S	N	N	N	N

En el par de reglas dependientes redundantes y mixtas 2'' y 3'' se puede eliminar la regla 2''b pues está contenida en la 3''b, y en el otro par 5'' y 6'', se puede eliminar la regla 5''b, quedando la tabla como se muestra:

¿Presenta otras enfermedades?	S	S	N	S	S	N
¿Condiciones Físicas Favorables?	S	N	-	S	N	-
¿Se conoce clasificación del Paciente?	S	S	S	N	N	N
Análisis de enfermedades que afectan el TR	x	-	-	x	-	-
Realizar análisis complementarios	-	x	x	-	x	x
Inserción en lista de espera	x	x	x	-	-	-
Remitir consulta de progresión	-	-	-	x	x	x
	1''	2''	3''	4''	5''	6''

Aplicando de nuevo el *criterio sobre reglas independientes y dependientes*:

1'' y 2'': son independientes

3'' y 4'': son independientes

1'' y 3'': son independientes

3'' y 5'': son independientes

1'' y 4'': son independientes

3'' y 6'': son independientes

1'' y 5'': son independientes

4'' y 5'': son independientes

1'' y 6'': son independientes

4'' y 6'': son independientes

2'' y 3'': son independientes

5'' y 6'': son independientes

2'' y 4'': son independientes

2'' y 5'': son independientes

2'' y 6'': son independientes

Todas las reglas ahora son independientes, por lo que el sistema de reglas integrado por estas reglas es ahora consistente:

¿Presenta otras enfermedades?	S	S	N	S	S	N
¿Condiciones Físicas Favorables?	S	N	-	S	N	-
¿Se conoce clasificación del Paciente?	S	S	S	N	N	N
Análisis de enfermedades que afectan el TR	x	-	-	x	-	-
Realizar análisis complementarios	-	x	x	-	x	x
Inserción en lista de espera	x	x	x	-	-	-
Remitir consulta de progresión	-	-	-	x	x	x
	1''	2''	3''	4''	5''	6''
	R1	R2'	R4	R5'	R6	R8

Las reglas iniciales, después de validadas, quedan como sigue: R1 se queda igual, R2 y R3 se validan quedando R2', R4 se queda igual, R5 y R7 se validan quedando R5', R6 y R8 se quedan igual.

R2’: Si el paciente presenta otras enfermedades, no tiene condiciones físicas favorables y se conoce su clasificación en consulta de nefrología se le realiza análisis complementarios y se inserta en la lista de espera.

R5’: Si el paciente presenta otras enfermedades, tiene condiciones físicas favorables 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.

3.4 Corrida del algoritmo implementado

En la presente investigación se ha programado una herramienta que detecta determinados tipos de anomalías como son: la ambivalencia de columna, columnas duplicadas y fila de acción irrelevante.

Paso 1: Editar las reglas en un documento *.txt a partir del enunciado.

La herramienta carga desde un fichero *.txt las reglas escritas con el formato siguiente:

```
If cond1,cond2,...,condN then accion1,accion2,...,accionN
```

Si en una de las reglas no se hace referencia al valor de una condición significa que dicha condición en esa regla toma el valor de “N”, y si se le pone delante a la condición el símbolo “\$” significa que no importa el valor que tome dicha condición, es decir, una indiferencia (-).

Veremos a través de varios ejemplos como realiza la validación del repositorio de RN.

Ejemplo 1:

R1: 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.

R9: Si el paciente presenta otras enfermedades, no tiene condiciones físicas favorables y se conoce su clasificación en consulta de nefrología se le realiza análisis de enfermedades que afectan el TR y se inserta en la lista de espera.

R2: Si el paciente no presenta otras enfermedades, no tiene condiciones físicas favorables y se conoce su clasificación en consulta de nefrología se le realiza análisis complementarios y se inserta en la lista de espera.

R4: Si el paciente no presenta otras enfermedades, tiene condiciones físicas favorables y se conoce su clasificación en consulta de nefrología se le realiza análisis de enfermedades de enfermedades que afectan el TR y se remite a consulta de progresión.

R5: Si del paciente se conoce su clasificación en consulta de nefrología se remite inmediatamente a consulta de progresión.

R6: Si el paciente no presenta otras enfermedades y tiene condiciones físicas favorables se le realiza análisis complementario.

R7: Si el paciente no tiene condiciones físicas favorables y no se conoce su clasificación en consulta de nefrología se le realiza análisis complementarios.

R8: Si el paciente no presenta otras enfermedades y no se conoce su clasificación en consulta de nefrología se le realiza análisis complementarios.

La matriz quedaría como sigue:

¿Presenta otras enfermedades?	S	S	N	N	-	N	-	N
¿Condiciones Físicas Favorables?	S	N	N	S	-	S	N	-
¿Se conoce clasificación del Paciente?	S	S	S	S	S	-	N	N
Análisis de enfermedades que afectan el TR	x	x	-	x	-	-	-	-
Realizar análisis complementarios	-	-	x	-	-	x	x	x
Inserción en lista de espera	x	x	x	-	x	-	-	-
Remitir consulta de progresión	-	-	-	x	x	-	-	-
	1	2	3	4	5	6	7	8

Se busca el fichero donde se tienen editadas las reglas con el formato antes descrito dando clic en el botón seleccionado en la figura 3.3:

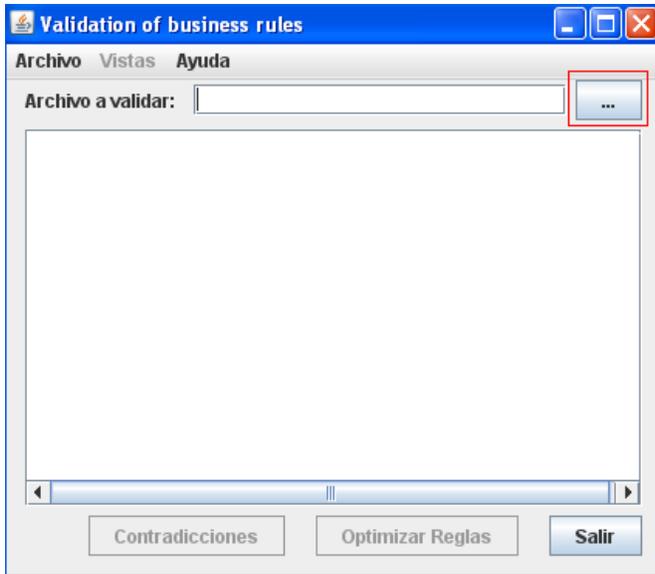


Figura 3.3: Ventana principal del validador de RN

Paso 2: Captar el conjunto de reglas a partir del documento *.txt y construir la tabla de decisión.

Luego se muestra una nueva ventana para seleccionar el fichero donde se encuentran las reglas editadas y se da clic en el botón “Abrir” que se muestra en la figura 3.4:

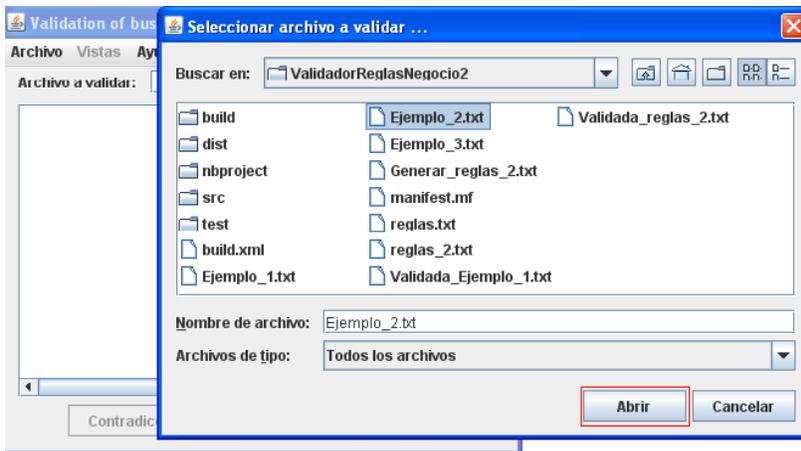


Figura 3.4: Ventana de selección del fichero de las RN editadas

Internamente en la aplicación se construye la tabla de decisión y luego se aplica el paso 3.

Paso 3: Verificar si existen ambivalencias o contradicciones entre las reglas.

Seguidamente se muestra la ventana principal nuevamente para realizar la validación se presiona el botón “Contradicciones”, como se muestra en la figura 3.5.

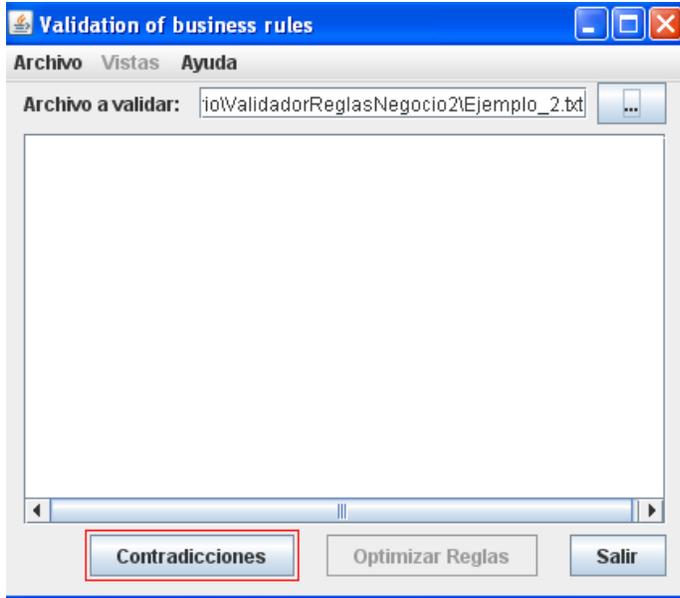


Figura 3.5: Ventana de chequeo de las contradicciones

Luego se muestra el resultado de dicho chequeo en una ventana de mensaje mostrando en el editor de la ventana las reglas, si existen anomalías del tipo de ambivalencia de fila o columna, se le hace saber al usuario entre qué reglas se presentó y el sistema no puede eliminar dicha anomalía pues es resultado de errores de construcción de la tabla de decisión por parte del usuario. En este ejemplo se evidencia este tipo de anomalía, mostrado en la figura 3.6.

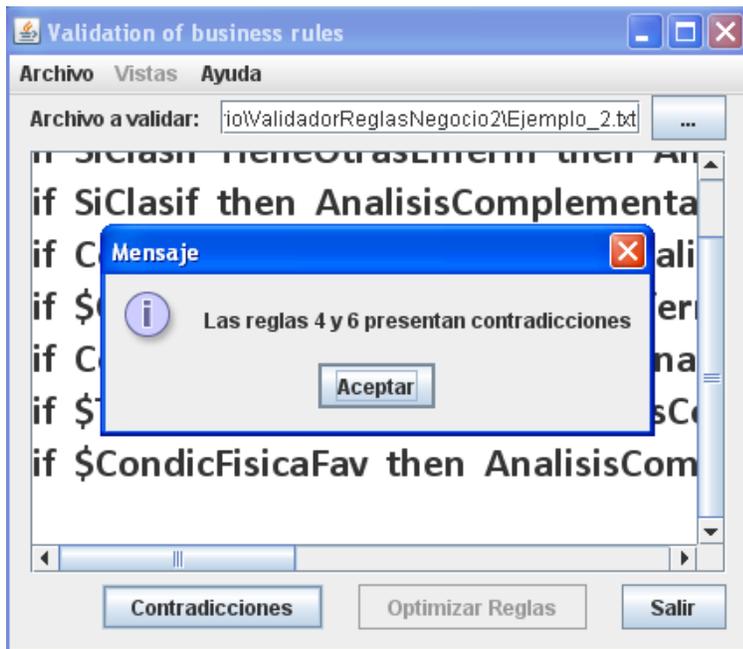


Figura 3.6: Ventana resultado del chequeo de anomalías

Lo que se recomienda en este caso es volver a editar las RN corrigiendo dicho error en las reglas 4 y 6 como se muestra en el mensaje de la figura 3.6, y luego volver a realizar todo el proceso.

Segundo ejemplo analizado con el validador.

Ejemplo 2:

R1: Si el paciente presenta otras enfermedades, tiene condiciones físicas favorables y se conoce su clasificación en consulta de nefrología se le realiza análisis de enfermedades que afectan el TR y se inserta en la lista de espera.

R2: Si el paciente presenta otras enfermedades, no tiene condiciones físicas favorables y se conoce su clasificación en consulta de nefrología se le realiza análisis de enfermedades que afectan el TR y se inserta en la lista de espera.

R3: Si el paciente no presenta otras enfermedades, no tiene condiciones físicas favorables y se conoce su clasificación en consulta de nefrología se le realiza análisis complementarios y se inserta en la lista de espera.

Paso 1: Editar las reglas en un documento *.txt a partir del enunciado.

if TieneOtrasEnferm,CondicFisicaFav,SiClasif **then**

AnalisisEnfermAfectanTR,InsertaListaEspera

if TieneOtrasEnferm,SiClasif **then**

AnalisisEnfermAfectanTR,InsertaListaEspera

if SiClasif **then**

AnalisisComplementarios,InsertaListaEspera

Paso 2: Captar el conjunto de reglas a partir del documento *.txt y construir la tabla de decisión.

La tabla de decisión correspondiente a este ejemplo es la siguiente:

¿Presenta otras enfermedades?	S	S	N
¿Condiciones Físicas Favorables?	S	N	N
¿Se conoce clasificación del Paciente?	S	S	S

Análisis de enfermedades que afectan el TR	x	x	-
Realizar análisis complementarios	-	-	x
Inserción en lista de espera	x	x	x
	1	2	3

Se busca el fichero donde están editadas las RN igual que como se hizo en el ejemplo anterior. El fichero es Ejemplo_1.txt.

Paso 3: Verificar si existen ambivalencias o contradicciones entre las reglas.

Para verificar si existen ambivalencia o contradicciones entre las reglas se da clic en el botón “Contradicciones”, ver figura 3.7.

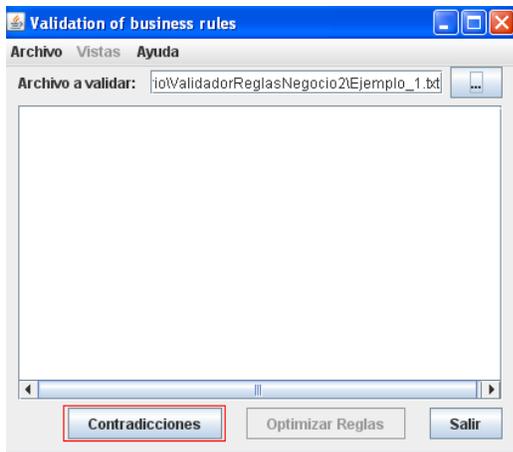


Figura 3.7: Ventana de chequeo de las contradicciones

Luego se muestra el resultado de dicho chequeo en una ventana de mensaje mostrando en el editor de la ventana las reglas, mostrado en la figura 3.8.



Figura 3.8: Ventana resultante del chequeo de anomalías

El mensaje muestra de que no presenta anomalías del tipo ambivalencia, que son las contradicciones, cuando se da clic sobre el botón “Aceptar” del mensaje, este se cierra y se muestra la ventana principal, ver figura 3.9.

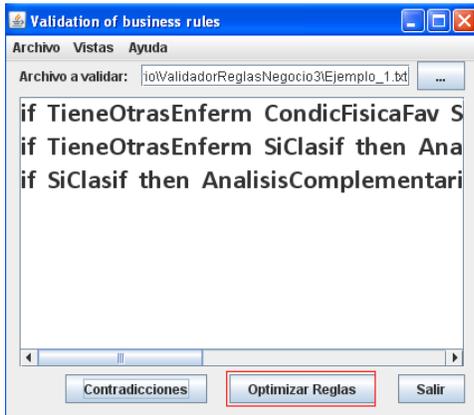


Figura 3.9: Ventana para optimizar las RN

Luego se da clic en el botón “Optimizar reglas” para eliminar posibles anomalías del tipo redundancia por columnas duplicadas, columnas subsumidas o fila de condición redundante, como se señala en la figura 3.9 en un cuadro de color rojo.

Durante la optimización de las reglas se realizan los pasos del 4 al 6 descritos en el epígrafe 2.5:

Paso 4: Aplicar primera ley: unificación de reglas por indiferencia.

Paso 5: Aplicar criterio de clasificación de pares de reglas.

Paso 6: Aplicar segunda o tercera ley de depuración.

Seguidamente se muestra la ventana de resultado de optimizar el repositorio de RN, ver figura 3.10, dando lugar a la ejecución del paso 7 y último del algoritmo.

Paso 7: Salida de reglas validadas en un fichero *.txt.

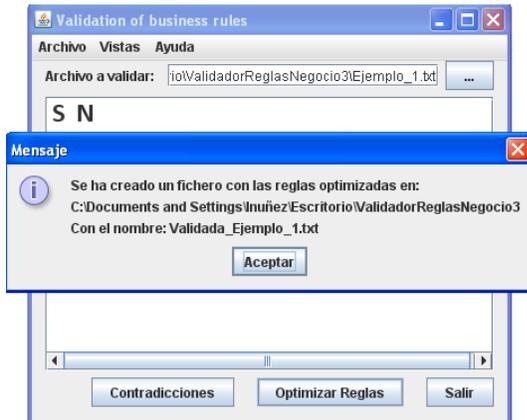


Figura 3.10: Repositorio de RN validado y optimizado

El mensaje muestra que se generó un fichero nombrado Validada_Ejemplo_1.txt con el conjunto de RN validado, cuando se da clic sobre el botón “Aceptar” del mensaje, en el editor de la ventana principal se muestra la matriz de condiciones y acciones, como resultado de la validación, ver figura 3.11.

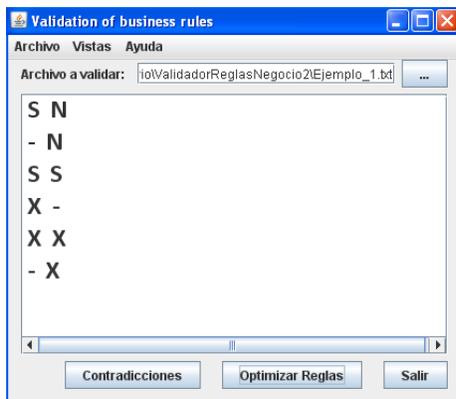


Figura 3.11: Repositorio de RN validado y optimizado

Como resultado de la validación y optimización del repositorio, se elimina la regla R2 quedando el repositorio como sigue:

R1: Si el paciente presenta otras enfermedades y se conoce su clasificación en consulta de nefrología se le realiza análisis de enfermedades que afectan el TR y se inserta en la lista de espera.

R3: Si el paciente no presenta otras enfermedades, no tiene condiciones físicas favorables y se conoce su clasificación en consulta de nefrología se le realiza análisis complementarios y se inserta en la lista de espera.

3.5 Conclusiones parciales del capítulo

En este capítulo se realiza la aplicación del algoritmo de validación de RN 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. Se implementa el algoritmo logrando identificar anomalías tales como ambivalencias y redundancias por columnas duplicadas, columnas subsumidas o fila de condición redundante.

Conclusiones

Como resultado de esta investigación se concluyó que es posible confeccionar un algoritmo basado en tablas de decisión que realice validaciones de la consistencia y chequeos de integridad de un conjunto de RN para eliminar inconsistencias y ambigüedades en el mismo, cumpliendo de esta forma con la hipótesis de investigación planteada.

1. Se realizó un estudio sobre métodos de validación de la consistencia para reglas de negocio basado en árboles de decisión, español estructurado y tablas de decisión.
2. Se valoraron los diferentes métodos estudiados para realizar la validación, centrando la presente investigación en tablas de decisión.
3. Se detallan los tipos de anomalías que pueden ser detectadas en el conjunto de reglas de negocio a partir de las tablas de decisión.
4. Se propone una vía o algoritmo para realizar la validación de la consistencia del conjunto de reglas de negocio basado en tablas de decisión, y se describen los pasos del mismo en un lenguaje de pseudocódigo.
5. Se aplica la vía o algoritmo al conjunto de reglas de negocio del caso de estudio del área de nefrología y se ejemplifica con reglas reales existentes.
6. Se confecciona una herramienta que detecta anomalías del tipo ambivalencia, redundancia por columnas duplicadas, columnas subsumidas o fila de condición redundante.

Recomendaciones

1. Valorar criterios para realizar validación inter-tabular y añadirla al algoritmo.
2. Ampliar la implementación de la herramienta de forma que sea posible identificar otros tipos de anomalías consideradas en la investigación, tales como: circularidad, deficiencias por reglas aisladas, columnas vacías; redundancia por fila de acción inusual, entre otras.
3. Dar continuidad al estudio con otros métodos para la validación de la consistencia del conjunto de RN.

Bibliografía

- AKAL, F., TÜRKER, C., SCHEK, H.-J., BREITBART, Y., GRABS, T. & VEEN, L. Fine-grained lazy replication and scheduling with freshness and correctness guarantees. Proceedings of the 31st VLDB Conference, 2005 Trondheim, Norway. Swiss Federal Institute of Technology Zurich, 565-576.
- AMAOLO, M. P. 2007. *Lenguajes de Modelado de Reglas de Negocio y la Web Semántica*. 1400, (8300) Neuquén, Universidad Nacional del Comahue.
- ANTONIOU, G., TAVETER, K., BERNDTSSON, M., WAGNER, G. & SPREEUWENBERG, S. 2004. A First-Version Visual Rule Language. *REWERSE, Report IST-*.
- APPELTON, D. S. 1984a. "Business Rules - The Missing Link". *Datamation*, Vol.30, pp. 145-150.
- APPELTON, D. S. 1984b. Business Rules - The Missing Link. *Datamation*, 30, Nro 16, 145-150.
- BAILEY, J., BRY, F. & P.-L., P. 2005. Composite event queries for reactivity on the web. *ACM Press [Online]*.
- BAILEY, J. B., F.; PÁTRÁNJAN P.-L. . 2005. Composite event queries for reactivity on the web. *ACM Press [Online]*.
- BAJEC, M. & KRISPER, M. 2005. A methodology and tool support for managing business rules in organisations. *Information Systems*, 30, 423-443.
- BAJEC, M., KRISPER, M. & RUPNIK, R. 2000. Using Business Rules Technologies To Bridge The Gap Between Business and Business Applications. In: RECHNU, G. E. (ed.) *Proc. of the IFIP 16th World Computer Congress 2000, Information Technology for Business Management*. Peking, China.
- BAJEC, M. & MARJAN, K. 2006. *Managing business rules in enterprises*. University of Ljubljana.
- BAJEC, M. A. K. M. 2006. *Managing business rules in enterprises*. University of Ljubljana: Ljubljana.
- BAJEC, M. A. M., K. 2001. Managing business rules in enterprises. *Electrotechnical Review, Ljubljana, Slovenija Elektrotehniški vestnik* 68(4) P. 236-241.
- BAJEC, M. K., M. ; RUPNIK, R. 2000. Using Business Rules Technologies To Bridge The Gap Between Business and Business Applications. In: RECHNU, G. E. (ed.) *Proc. of the IFIP 16th World Computer Congress 2000, Information Technology for Business Management*. Peking, China.
- BARNE, M. & KELLY, D. 1997. Play by the Rules. *Byte (Special Report)*, 22, Nro 6, 98-102.
- BARNE, M. A. D. K. 1997. Play by the Rules. *Byte (Special Report)*, p. 98-102.
- BOGGIANO CASTILLO, M. B., MARÍNEZ DEL BUSTO, M. E., PÉREZ ALONSO, A. & PÉREZ VÁZQUEZ , R. Aplicación de Reglas de Negocio mediante triggers. CIE 2007, XIII Convención de Ingeniería Electrónica, 20-22 de Julio 2007a Santa Clara, Villa Clara. Universidad Central de Las Villas.
- BOGGIANO CASTILLO, M. B., MARTÍNEZ DEL BUSTO, M. E., PÉREZ ALONSO, A., GONZÁLEZ GONZÁLEZ , L. & PÉREZ VÁZQUEZ , R. Implementación de reglas de negocio de restricciones y patrones de sentencias de restricciones con triggers. UCIencia 2007, 2007b Cuba.
- BOGGIANO CASTILLO, M. B., PERÉZ ALONSO, A., MARTÍNEZ DE BUSTO, M. E. & PERÉZ VAZQUEZ, R. Enfoque de reglas de negocio asociadas a la inteligencia organizacional y su automatización en bases de datos. Eureka, 2010 Ciudad de la Habana, Cuba.
- BOGGIANO CASTILLO, M. B., PERÉZ ALONSO, A., MARTÍNEZ DE BUSTO, M. E., PERÉZ VAZQUEZ, R. & GONZÁLEZ GONZÁLEZ , L. 2009a. GENERATING RESTRICTION RULES AUTOMATICALLY WITH AN INFORMATION SYSTEM. *Revista Cubana Informática Médica*. CECAM.
- BOGGIANO CASTILLO, M. B., PERÉZ ALONSO, A., PERÉZ VÁZQUEZ, R., MARTÍNEZ DE BUSTO, M. E. & GONZÁLEZ GONZÁLEZ , L. Traducción de un Patrón de Reglas de Negocio en Bases de Datos Relacionales. Seventh LACCEI Latin American and Caribbean Conference for Engineering and Technology (LACCEI'2009), "Energy and Technology for the Americas: Education, Innovation, Technology and Practice", June 2-5, 2009 2009b San Cristóbal, Venezuela.
- BRG 2000a. Defining Business Rules ~ What Are They Really?
- BRG 2000b. Defining Business Rules ~ What Are They Really? GUIDE Business Rules Project Final Report

- In: VESIÓN 1.3, J. (ed.). Seattle: Business Rules Group.
- BRG 2003a. *Manifiesto de Reglas de Negocio: Los Principios de la Independencia de las Reglas*.
- BRG 2003b. *Manifiesto de Reglas de Negocio: Los Principios de la Independencia de las Reglas*.
- BRG 2009. Business Rule Group, a non-commercial peer group of IT professionals.
- CASTILLA, M. J. 2007. Tablas de decisión en sistemas de información II. 1. Available: <http://www.facso.unsj.edu.ar/catedras/ciencias-economicas/sistemas-de-informacion-II/documentos/tabla.pdf> [Accessed January 18, 2012].
- CASTILLA, M. J. 2008. Cursogramas en Sistemas de Información II. *Cursogramas* [Online]. Available: http://www.fchst.unlpam.edu.ar/cartelera/curso_extra/cursogramas_mjcastilla.pdf.
- CERI, S. & FRATERNALE, P. 1997a. "Designing Database Applications with Objects and Rules: The IDEA Methodology", Addison-Wesley.
- CERI, S. & FRATERNALE, P. 1997b. Designing Database Applications with Objects and Rules: The IDEA Methodology. Addison-Wesley.
- CONTRERAS BAEZ, C. & BOLANOS RODRÍGUEZ, H. A. 2011. Método para la extracción de reglas de negocio aplicados a casos de uso en procesos empresariales. *Universidad de San Buenaventura, Facultad de Ingeniería, Especialización de procesos para el desarrollo de software, Santiago de Cali*.
- CHANG, V. N. 2004. Método de Dos Fases para la Asignación de Datos en una Base de Datos Distribuida. Available: <http://www.informatizate.net> [Accessed 27 de octubre de 2004].
- CHAPPEL, O. 2005. Term-Fact Modeling, the Key to Successful Rule-Based Systems. *Business Rules Journal*, Vol. 6.
- DATE, C. J. 2000a. What Not How: The Business Rules Approach To Application Development. Addison Wesley Longman,.
- DATE, C. J. 2000b. What Not How: The Business Rules Approach To Application Development. Addison Wesley Longman Inc.
- DOBSON, J. 1992. *A Methodology for Managing Organisational Requirements*. University of Newcastle upon Tyne.
- FALQUET, G., GUYOT, J., JUNET, M., LÉONARD, M., BURSENS, R., CRAUSAZ, P. & PRINCE, I. Concept Integration as an Approach to Information Systems Design. In: BHABUTA, A. A. V.-S. A. L., ed. Computerized Assistance During the Information Systems Life Cycle, 1988 North-Holland, Amsterdam T.W. Olle, p.19 - 65.
- GOEDERTIER, S., HAESSEN, R. & VAN THIENEN, J. 2007a. EM-Bra2CE v0.1: A Vocabulary and Execution Model for Declarative Process Models. *FETEW research report*. K.U.Leuven.
- GOEDERTIER, S., MUES, C. & VAN THIENEN, J. 2007b. Specifying Process-Aware Access Control Rules in SBVR.
- GOEDERTIER, S. & VAN THIENEN, J. Rule-based business process modeling and execution. In: DEPARTMENT OF DECISION SCIENCES & INFORMATION MANAGEMENT, N.-L., ed. EDO 2005, VORTE 2005, 2005 Leuven. Katholieke Universiteit
- GOEDERTIER, S. & VAN THIENEN, J. 2006a. Rule-based business process modeling and execution, in Department of Decision Sciences & Information Management Naamsestraat 69-3000 Leuven. *Katholieke Universiteit Leuven*.
- GOEDERTIER, S. & VAN THIENEN, J. 2006b. *Rule-based business process modeling and execution*. Katholieke Universiteit
- GOTTESDIENER, E. 1999. "Capturing Business Rules". *Software Develop. Mag., Manage. Forum*.
- HALPIN, T. 2000a. *A Fact-Oriented Approach to Business Rules*, ER.
- HALPIN, T. 2007. Fact-Oriented Modeling: Past, Present and Future. *Conceptual Modelling in Information Systems Engineering*, Springer-Verlag Berlin Heidelberg.

- HALPIN, T. A. 2000b. A Fact-Oriented Approach to Business Rules. *ER*, 582-583.
- HAMADI, R. & BENATALLAH, B. A Petri net-based model for web service composition,. the Fourteenth Australasian database conference on Database technologies 2003, 2003 Darlinghurst, Australia, Australia. Australian Computer Society, 191-200.
- HAY, D. 1997. GUIDE Business Rules Project, Fial Report - revision 1.2. *GUIDE International Corporation*. Chicago.
- HAY, D. & HEALY, K. A. 1997. "GUIDE Business Rules Project, Final Report - revision 1.2". *GUIDE International Corporation, Chicago*.
- HENDRYX, S. 2003. Business Rules and Standards. *A Fair Isaac White Paper*.
- HERBST, H. 1995. *A Meta-Model for Business Rules in Systems Analysis*.
- HERBST, H., KNOLMAYER, G., MYRACH, T. & SCHLESINGER, M. 1994a. "The Specification of Business Rules: A Comprison of Selected Methodologies", *Methods and Associated Tools for the Information Systems Life Cycle*, (A. Verrijin and T. W. Olle, Ed), Amsterdam at al.: Elsevier 1994.
- HERBST, H., KNOLMAYER, G., MYRACH, T. & SCHLESINGER, M. 1994b. *The Specification of Business Rules: A Comprison of Selected Methodologies* *Methods and Associated Tools for the Information Systems Life Cycle*, Amsterdam at al., A. Verrijin and T. W. Olle: .
- HÜSEMANN, S. & SCHÄFER, M. Building flexible eHealth processes using business rules. European Conference on eHealth 2006, 2006 Switzerland. Lecture Notes in Informatics (LNI).
- KASISKE, B., CANGRO, C., HARIHANAN, S., HRICIK, D., KERMAN, R., ROTH, D., RUSH, D., VAZQUEZ, M. & WEIR, M. 1995. The evaluation of renal transplant candidates: Clinical practice guidelines. *J Am Transplant 2(suppl 1)*, 5-95.
- KELLER, G., NUTTGENS, M. & SCHEER, A. 1992. *Semantische Prozessmodellierung auf der Grundlage Ereignisgesteuerter Prozessketten (EPK)*. Institut f`ur Wirtschaftsinformatik Saarbr`ucken.
- KOVACIS, A., GROZNIK, A. & KRISPER, M. 2002. Business renovatios: from business process modelling to infomation system modelling. *I. J. of SIMULATION* [Online], Vol. 2 No. 2
- LINEHAN, M. H. & DE SAINTE MARIE, C. 2011. Semantics of Business Vocabulary & Business Rules (SBVR): The Relationship of Decision Model and Notation (DMN) to SBVR and BPMN.
- LIPECK, U. W. (ed.) 1989. *Dynamische Integrität von Datenbanken*, Berlin Springer.
- LÓPEZ G., E. A. F. J. I. 2006. Una propuesta de Modelos de Ciclo de Vida (MCVS) para la Integración de los Procesos de Negocio utilizando Service Oriented Architecture (SOA). *Departamento de Estudios e Investigación de la Empresa-Facultad de Economía Italia, Universidad de Salerno*.
- MARISEL MASS CHAVIANO & GARCÍA, M. M. 2010. Editor de Reglas de Negocio basado en un Modelo de Hechos para una Claificación Semántica. *Universidad Central "Marta Abreu" de Las Villas*.
- MARTÍNEZ BUSTO, M. E. 2011. *Captación de reglas de negocio para el desarrollo de sistemas de información sobre un dominio específico*. Tesis Doctoral, Universidad Central "Marta Abreu" de Las Villas.
- MARTÍNEZ BUSTO, M. E., MENDILAHAXON VALDÉS, L. F., CUELLAR VEGA, E. & GONZÁLEZ GONZÁLEZ, L. Modelación de reglas y procesos para la prescripción medica y el control de medicamentos en farmacia intrahospitalaria. Seventh LACCEI Latin American and Caribbean Conference for Engineering and Technology (LACCEI'2009), "Energy and Technology for the Americas: Education, Innovation, Technology and Practice", June 2-5, 2009 2009 San Cristóbal, Venezuela.
- MARTÍNEZ BUSTO, M. E., MENDILAHAXON VALDÉS, L. F., PHAN, A. D. & NÚÑEZ PÉREZ, L. Sistema para el Control Intrahospitalario de Medicamentos. VII Conferencia Científica Internacional UNICA 2006, 2006 Cuba.
- MARTÍNEZ BUSTO, M. E., MONTES DE OCA MORENO, I., BISHOKSA, K., CABALLERO MARTÍNEZ, A., BOGGIANO CASTILLO, M. B., PERÉZ ALONSO, A. & GONZÁLEZ GONZÁLEZ, L. AUTHORIZING,

- MANAGING AND IMPLEMENTING BUSINESS RULES RELATED TO KIDNEY TRANSPLANTATION USING DROOLSTAB PLUG-IN FOR PROTÉGÉ. Workshop 2010, 2010a Santa Clara, Cuba.
- MARTÍNEZ BUSTO, M. E., MORENO MONTES DE OCA, I., RODRÍGUEZ MORFFI, A., CASTRO ARTILES, M. & GONZÁLEZ GONZÁLEZ, L. 2010b. Vocabulario de negocio para trasplante renal con enfoque ontológico para un modelo de hechos genérico. *Revista de la Facultad de Ingeniería, Universidad de Antioquía*, N.º 53 pp. 155-162.
- MARTÍNEZ FERNÁNDEZ, J. L. 2010. *Introduciendo semántica en un proceso de desarrollo software a través de reglas de negocio. Tesis doctoral.*, Universidad Politécnica de Madrid.
- MASS CHAVIANO, M. & MARQUEZ GARCIA, M. 2010. *Editor de Reglas de Negocio basado en un Modelo de Hechos para una Clasificación Semántica*. Trabajo de Diploma, Universidad Central de las Villas.
- MEREDITH, L. G. & BJORG, S. 2003. Contracts and types. *Commun. ACM*, 46, no. 10, 41-47.
- MICROSOFT. 2003. *OWL-S: Semantic Markup for Web Services* [Online]. Microsoft Press. Available: <http://www.daml.org/services/owl-s/>.
- MORENO MONTES DE OCA, I. 2008. *Representación del Modelo de Hechos Mediante Ontologías*. Tesis presentada en opción al Título Académico de Master en Computación Aplicada Universidad Central de Las Villas.
- MORENO MONTES DE OCA, I., GONZÁLEZ GONZÁLEZ, L. & MARTÍNEZ DEL BUSTO, M. E. 2008. *Modelo de hechos genérico*. Cuba patent application.
- MORGAN, T. 2002. *Business Rules and Information Systems: Aligning IT with Business Goals*, Addison Wesley.
- MORIARTY, T. 2000. Business Rule Management Facility: System Architect 2001. *Intelligent Enterprise*, 3, No. 12.
- NILSSON, B. E. (ed.) 1999. *"On why to model what and how: concepts and architecture for change, Perspectives on Business Modelling—Understanding and Changing Organisations"* New York: Springer.
- NÚÑEZ PÉREZ, L., MARTÍNEZ BUSTO, M. E., GONZÁLEZ GONZÁLEZ, L., HERNÁNDEZ HERNÁNDEZ, P. & PÉREZ VÁZQUEZ, R. 2010. Clasificación semántica de reglas de negocio, patrones de diseño y validación. *Memorias de la VII Conferencia Internacional de Ciencias Empresariales. Hotel Melía Las Dunas. Destino turístico "Cayos de Villa Clara", Cuba del 15 al 17 de Octubre*.
- O'KEEFE, R. M. & O'LEARY, D. E. 1993. Expert system verification and validation: a survey and tutorial. *Artificial Intelligence Review* 7.
- OJO, A., HANSON, J., HU., M.-K., OKECHUKWU, C., WOLFE, R., LIECHTMAN, A., AGODOA, L., KAPLAN, B. & PORT, F. 2001. Survival in recipients of marginal cadaveric donor kidneys, compared to other recipients and wait-listed transplant patients. *J Am Soc Nephrol* 12, 589-597.
- OMG 2008. *Semantics of Business Vocabulary and Business Rules (SBVR)*, v 1.0.
- PETROUNIAS, I. & LOUCOPOULOS, P. A rule based approach for the design and implementation of information systems. In: JARKE, M., ed. *Proceedings EDBT '94*, 1994 Cambridge, UK. Springer.
- PREECE, A. & SHINGHAL, R. 1994. Foundation and application of knowledge base verification. *Int. Journal of Intelligent Systems*, 9, 683-701.
- ROSCA, D., GREENSPAN, S., FEBLOWITZ, M. & WILD, C. A. 1997a. Decision making methodology in support of the business rules lifecycle. *Proc. of the 3rd IEEE International Symposium on Requirements Engineering*.

- ROSCA, D., GREENSPAN, S., FEBLOWITZ, M. & WILD, C. A. "Decision making methodology in support of the business rules lifecycle". Proc. of the 3rd IEEE International Symposium on Requirements Engineering, 1997b. pp. 236 -246.
- ROSCA, D., GREENSPAN, S., WILD, C., REUBEINSTEIN, H., MALY, K. & FEBLOWITZ, M. Application of a decision support mechanism to the business rules life cycle. Proceedings of the 10th, 1995.
- ROSCA, D. G., S.; FEBLOWITZ, M.; WILD, C. A. 1997. "Decision making methodology in support of the business rules lifecycle". *Proc. of the 3rd IEEE International Symposium on Requirements Engineering* [Online].
- ROSS, R. G. 1997a. "The Business Rule Book: Classifying, Defining and Modelling Rules". *Business Rule Solutions, Inc.*
- ROSS, R. G. 1997b. *The Business Rule Book: Classifying, Defining and Modelling Rules*, Texas, Huston, Ross Method, version 4.0.
- ROSS, R. G. 2000. What are Fact Models and why do you need them? . *Business Rules Journal*.
- ROSS, R. G. 2003. *Principles of the Business Rule Approach*, Addison-Wesley.
- ROSS, R. G. 2005. *Business Rule Concepts*.
- ROSS, R. G. 2006. Business Rules Concepts-Getting to the Point Knowledge.
- ROSS, R. G. 2008. Business Rules from A –Z:What You Need to Know Software Engineering der Schweizer Informatik Gesellschaft. *Business Rules Journal*.
- ROSS, R. G. & LAM, G. S. W. 2003. Developing the Business Model The Steps of Business Rule Methodology.
- ROSS, R. G. & LAM, G. S. W. 2006. Developing the Business Model The Steps of Business Rule Methodology.
- SBVR A Brief History of the Business Rule Approach. *Business Rules Journal*.
- SCHEWE, K.-D. On the unification of query algebras and their extension to rational tree structures. *In: ORLOWSKA, M. E. & RODDICK, J., eds. Proceedings of the Australasian Database Conference (ADC2001), 2001. Australian Computer Society, Inc.*
- SOLIVERES, P. A. 1997. Desarrollo Cliente/Servidor: ubicación de las reglas de negocio. *Revista Profesional para Programadores (RPP)* [Online].
- SPREEUWENBERG, S. & GERRITS, R. Business Rules in the Semantic Web, are there any or are they different? Reasoning Web: Second International Summer School, 2006.
- STRUCK, D. L. 1999a. "Business Rule Continuous Requirements Environment". PhD Thesis PhD Thesis, Colorado Technical University.
- STRUCK, D. L. 1999b. *Business Rule Continuous Requirements Environment*. Colorado Technical University.
- STRUCK, D. L. 1999c. *Business Rule Continuous Requirements Environment*. Colorado Technical University.
- TAVETER, K. & WAGNER, G. 2001. *Agent-Oriented Business Rules: Deontic Assignments*. VTT Information Technology, Eindhoven Univ. of Technology, Fac. of Technology Management.
- URBAN, S. D., KARADIMCE, A. P. & NANNAPANENI, R. B. The Implementation and Evaluation of Integrity Maintenance Rules in an Object-Oriented Database. *In: SOCIETY, I. C., ed. Proceedings Eighth International Conference on Data Engineering, 1992 Los Alamitos IEEE Computer Society Press, p. 565 - 572.*
- USCHOLD, M. & GRUNINGER., M. 1996. "ONTOLOGIES: Principles, Methods and Applications". *Knowledge Engineering Review*, Volume 11
- VALIENTE, G. 1991. Are Complete K-Trees Something More Than Decision Trees. *Sigar Bulletin. ACM Press*.

- VAN DER AALST, W. M. P., TER HOFSTEDE, A. H. M., KIEPUSZEWSKI, B. & BARROS, A. P. 2003. Workflow Patterns. *Distrib. Parallel Databases*, 14, no. 1, 5-51.
- VANIACHINE, A., MALON, D. & VRANICAR, M. 2005. Advanced technologies for distributed database services hyperinfrastructure *International Journal of Modern Physics, A* 20, 3877-3879.
- VANTHIENEN, J. 2003. PROcedural LOGic Analyzer (PROLOGA) 5.2. User's Manual.
- VANTHIENEN, J. 2008. Tablas de Decisión y de Reglas de Negocio. Universidad Católica de Lovaina.
- VANTHIENEN, J., AERTS, A. & MUES, C. 1995. A modelling approach to KBS verification. *European Symposium on the Validation and Verification of Knowledge Based Systems (EUROVAV 95). Chambéry, France*, 155-171.
- VANTHIENEN, J., MUES, C. & AERTS, A. 2009. An Illustration of Verification and Validation in the Modelling Phase of KBS Development.
- VANTHIENEN, J. & WETS, G. 1995. Integration of the decision table formalism with a relational database environment. *Katholieke Universiteit Leuven, Department of Applied Economic Sciences, Naamsestraat 69, B-3000 Leuven, Belgium*.
- VASILECAS, O. & LEBEDYS, E. 2007. Application Of Business Rules For Data Validation. *Information Technology and Control*, 36, No. 3.
- VON HALLE, B. 1994. Back to Business Rule Basics. *Database Programming & Design*, pp. 15-18.
- VON HALLE, B. 2002. Business Rules Applied: Building Better Systems Using the Business Rule Approach. *Wiley Computer*.
- WEIDEN, M., HERMANS, L., SCHREIBER, G. & VAN DER ZEE, S. 2002. Classification and Representation of Business Rules.
- WEIDEN, M. R. 2000. *A Critique of the Business-Rule Approach*. University of Amsterdam.
- WEIDEN, M. R., HERMANS, L., SCHREIBER, G. & VAN DER ZEE, S. 2004. Classification and Representation of Business Rules., www.omg.org/docs/ad/02-12-18.pdf.
- WIDOM, J. & CERI, S. 1996. "Active Database Systems - Triggers and Rules For Advanced Database Processing", San Francisco, Morgan Kaufmann.
- WILKES, L. & VERYARD, R. 2004. Service-Oriented Architecture: Considerations for Agile Systems. *Microsoft Architects Journal*, April, 11-23.
- YOUDEOWEI, A. 1997. *The B-Rule Methodology: A Business Rule Approach to Information Systems Development*.
- ZACHMAN, J. 1987. A Framework for Information Systems Architecture. *IBM Systems Journal*,, 26 No. 3.
- ZACHMAN, J. A. 2002. The ZachmanFramework: A Primer for Enterprise Engineering and Manufacturing.

Anexos

Anexo 1: Clasificación de reglas de negocio según diferentes criterios

En este enfoque de RN, y en la etapa de modelación dentro del ciclo de vida de las reglas, se necesita clasificarlas para facilitar el trabajo.

(1) Clasificación respecto a la contraparte interna en el SI

Una de las primeras formas en que las de RN son clasificadas aparece en (Herbst et al., 1994b); donde los eventos, procesos y objetos del mundo real deben ser distinguidos desde su contraparte interna del SI. En un caso se establece la diferencia entre reglas de actividad y de consistencia. Las reglas de consistencia, a su vez, pueden ser clasificadas como: restricciones activas y pasivas (Urban et al., 1992).

Otra clasificación común distingue entre las restricciones de integridad las dinámicas y estáticas (Falquet et al., 1988), expandiéndose las estáticas como aquellas que restringen los estados de la BD a un subconjunto permitido con relación a estados previos; y las restricciones dinámicas o transicionales como aquellas que llevan a estados arbitrarios de la BD, según lo permitido (Lipeck, 1989).

Finalmente Herbst (Herbst et al., 1994b) resume estas ideas, clasifica las RN en: de actividad, aquellas que prescriben acciones que deben llevarse a cabo; y de consistencia, aquellas que definen restricciones de integridad sobre estados válidos de la base de datos.

Expandiendo estas últimas en restricciones de integridad pasiva y activa.

En virtud de lo anterior, se considera que el autor se enfoca en establecer una relación directa entre las RN y el SI. En este sentido se complejiza el descubrimiento y extracción de las reglas.

(2) Clasificación por su accionabilidad

Otra importante clasificación ha sido la propuesta por Taveter y Wagner (Taveter and Wagner, 2001), para estos autores las RN pueden ser clasificadas atendiendo a su posible control. Según señalan se dividen en dos grandes grupos: reglas operativas y estructurales.

Reglas operativas: aquellas que pueden ser infringidas por las personas, tienen un sentido de obligación o prohibición y pueden ser preventivas. Su incumplimiento provoca consecuencias en

el negocio. Este tipo de reglas comparten conductas del negocio, monitorean aspectos como iteraciones, acuerdos a nivel de servicio y conformidad.

Reglas estructurales: necesarias para evaluar las reglas operativas. Mediante estas reglas se establece cómo la empresa define su conocimiento básico, tienen sentido de necesidad o imposibilidad. Comparten el conocimiento del negocio y no se incumplen directamente por el hombre. Las reglas de esta categoría son más conceptuales, pero su diferencia esencial de las anteriores radica en que establecen los límites que permiten evaluar la validez de “algo”, indican la línea de demarcación y cuándo “algo” es o no instancia de un concepto. La definición de reglas estructurales expresa lo esencial del concepto, representando un resultado más estable.

Por definición, las reglas estructurales expresan lo esencial de los conceptos, resultan más estables y son las encargadas de extraer el conocimiento de dos formas básicas: por cómputo y mediante la toma de decisiones.

Según el propio autor, las RN se dividen en dos categorías: no automatizables, reglas relacionadas con el comportamiento de los actores del negocio; y automatizables, que rige el comportamiento del negocio desde el sistema que le da soporte al mismo.

De lo anterior puede deducirse que la taxonomía propuesta por este autor no ofrece claridad para identificar las reglas desde la visión del negocio por manejar criterios muy amplios para identificarlas.

(3) Clasificación según el BRG

Por su parte, el BRG plantea que cada RN debe estar en una de las categorías siguientes (BRG, 2000b):

Sentencias estructurales: son importantes para la propia existencia del negocio. Este puede ser un concepto de interés o una relación entre concepto. Detalla una especificación o un aspecto estático del negocio y expresa conocimiento de objetos. Generalmente aparecen en el modelo entidad/ interrelación.

Sentencias de acción: son restricciones o condiciones que limitan o controlan acciones de una empresa. Representan una sentencia relativa a algunos aspectos dinámicos del negocio. Mediante ellas se especifican restricciones sobre el resultado que una acción puede producir. Las

restricciones son descripciones no procedurales en términos de otras RN atómicas. Mientras que las sentencias estructurales describen posibilidades, las sentencias de acción imponen restricciones (puede o debe, no puede o no debe).

Derivación: Sentencia que conoce lo que se deriva desde otro conocimiento del negocio.

La clasificación propuesta por el BRG (BRG, 2000b) resulta muy concisa y esclarecedora en cuanto al límite entre las categorías, aunque no logra suficiente claridad para el manejo posterior de las RN.

(4) Clasificación por su funcionalidad

Según señalan Kovacic y otros autores (Kovacic et al., 2002) que las reglas pueden ser divididas en definición, restricción y derivación (véase la Figura A1.1).



Figura A1.1 Categorías de reglas de negocio (Kovacic et al., 2002)

Anexo 2: Pseudocódigo del algoritmo de validación de la consistencia de un conjunto de reglas

Basado en el método descrito en el epígrafe anterior, se ha realizado un pseudocódigo de cómo sería la programación del mismo.

```

programa Principal()
  var
    Condiciones, Acciones: Matriz
    F: Arreglo;
  begin
    Condiciones= IdentificarCondiciones();
    Acciones= IdentificarAcciones();
    PrimeraLey(var Condiciones, var Acciones);
    SegundaLey(var Condiciones, var Acciones,F);
  end
procedure PrimeraLey(var Condiciones:Matriz;var Acciones:Matriz);
  var
    a,b:Arreglo;
  begin
    a=AccionesIguales(Acciones);
    b=Indiferentes(Condiciones,a);
    (Acciones,Condiciones)=Unificar(Acciones,Condiciones,b);
  end

procedure SegundaLey(var Condiciones:Matriz;var Acciones:Matriz);
  var
    a,b:Arreglo;
  begin
    a=ReglasDependientes(Condiciones);
    if(ReglasContradictorias(Acciones,a))
      Imprimir("Error en el enunciado de las reglas o mal construccion de la tabla");
    else
      begin
        b=EliminarReglasPuras(Acciones,Condiciones,a);
        TerceraLey(Acciones,Condiciones,b);
      end
    end
  end

procedure TerceraLey(var Condiciones:Matriz;var Acciones:Matriz,F:Arreglo);
  var
    a,b:Arreglo;
    cond,acci:Matriz;
  begin

```

```
while(HallanReglasMixtasDependientes(F))
  begin
    a=TomarReglasMixtasDependientes(F);
    cond=DescomponerPuras(Condiciones,a);
    EliminarReglasRepetidas(cond);
    acci=CrearMatrizAcciones(cond.length());
    (acci;cond)=PrimeraLey(acci,cond);
    F=Modificar(F,a);
  (Acciones,Condiciones)=Modificar(Acciones,Condiciones,acci,cond);
  end
end
```