Universidad de las Ciencias Informáticas

Facultad 3

Título: Ontología del modelo multidimensional y las asociaciones entre sus elementos

Trabajo de Diploma para optar por el título de Ingeniero Informático

Autor: Guillermo Manuel Negrín Ortiz

Tutor: Julio César Díaz Vera

Co-tutor: Andy Fernández Garabote

Julio 2014

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Guillermo Manuel Negrín Ortiz	**Julio César Díaz Vera**
_____	_____
Firma del Autor	Firma del Tutor

Datos de Contacto

MSc. Julio Cesar Díaz Vera

Universidad Martha Abreu, Villa Clara, Cuba.

Correo electrónico: jcdiaz@uci.cu

Ing. Andy Fernández Garabote

Universidad de las Ciencias Informáticas, La Habana, Cuba.

Correo electrónico: agarabote@uci.cu

Dedicatoria

A mis padres,

Por su ejemplo de perseverancia y dedicación,

por su estimulante optimismo y confianza,

por su amor sin límites

Agradecimientos

Al profesor Julio César Díaz Vera, quien asumió la responsabilidad de tutorar este trabajo, el cual enriqueció con su vasta experiencia sin descuidar ningún detalle y quien ha dejado en mí no solo la motivación de profundizar en los conocimientos en los que se basó esta investigación, sino también el incentivo de continuar profundizando en el campo de las bases de datos, por lo que le estoy muy agradecido.

A los profesores miembros del tribunal, quienes tuvieron a su cargo la revisión de este trabajo el cual rectificaron paso a paso, aportándome sus experiencias y enseñanzas que me permitieron profundizar en los temas tratados, reconociendo así sus cualidades como especialistas en este campo de la investigación.

A todos los profesores de la carrera, sin excepción y en especial a la profesora Dariela, que de una forma u otra contribuyeron con sus enseñanzas a que pudiera culminar el anhelado sueño de hacerme profesional de las Ciencias Informáticas y en particular a los que me asistieron en el perfeccionamiento del informe final de mi tesis de grado.

A mis compañeros de estudio, que posiblemente sin que ellos mismos lo sepan, me incentivaron a muchas de las actividades realizadas en la docencia y la investigación durante la carrera, me sirvieron de apoyo en no pocas ocasiones y en los que vi reflejadas mis propias aspiraciones. Tendría que mencionar a muchos, pero creo que en estos momentos vale la pena agradecer muy especialmente a Rayner, Leonardo, Yoalvy, Nilson y Leandro.

A mi familia porque siempre me incentivó a alcanzar metas superiores en mi preparación como estudiante y como ser humano, por su comprensión y apoyo permanentes, pero por sobre todo, por el ejemplo de dedicación y sacrificio que

implican llegar a ser un profesional y el compromiso que asumo de ahora en adelante con la profesión que escogí, con mi país y con ellos mismos.

Guillermo Manuel Negrín Ortiz

Resumen

La no existencia de un mecanismo para representar de manera integrada la estructura de un cubo de datos, el conocimiento del dominio y las relaciones entre los elementos del mismo que pueda ser interpretado tanto por usuarios humanos como tecnológicos da origen a esta investigación, cuyo objetivo es desarrollar una ontología que permita darle solución a esta problemática. La propuesta se realiza valorando su aplicación a la inteligencia de negocios. Se abordan los conceptos fundamentales de almacenes de datos, haciendo énfasis en el modelo multidimensional y su estructura. Se describen las reglas de asociación como resultado del proceso de minería de datos y se explica el problema que representa su extracción y la redundancia. Se introducen elementos de conocimiento previo para analizar su efecto en la redundancia de las reglas de asociación y definir la forma en que se representa este. Se definen una ontología “core” y puntos de extensión para construir modelos de dominio a partir de información disponible en cubos de datos. Se seleccionan los elementos del modelo multidimensional para representarlos como parte de la ontología**,** codificados en el lenguaje OWL2. Se aplica una variante de evaluación que establece cuatro etapas, además se ejemplifica el proceso utilizando dos cubos de datos. Se constató la validez de la ontología para extenderla a la representación ontológica de un dominio, demostrándose el correcto funcionamiento de los mecanismos de transformación.

PALABRAS CLAVE

Ontología, modelo multidimensional, ontología core, ontología de dominio, cubo de datos, reglas de asociación.

Tabla de contenidos

Contenido

Introducción 1
Problema a resolver 4
Objetivo General 4
Idea a defender 4
Objeto de Estudio 4
Campo de Acción: 5
Posibles resultados 5
Objetivos Específicos 5
Tareas a cumplir 5
Descripción de los capítulos 6
Capítulo 1 7
Fundamentación Teórica 7
Introducción 7
1.1 Modelo multidimensional y OLAP 7
1.1.1 Almacenes de datos 7
1.1.2 OLAP y OLTP 9
1.1.3 Modelo Multidimensional 10
1.2 Minería de datos 13
1.2.1 Knowledge Discovery in Databases 13
1.2.2 Reglas de asociación 15
1.2.3 Soporte y confianza 15
1.2.4 Extracción de reglas y Algoritmo de extracción 16
1.2.5 Ejemplo de reglas generadas 19
1.3 Ontologías 21
1.3.1 Conocimiento previo 21
1.3.2 Definición de ontología 22
1.3.3 Lenguajes para representar ontologías 24
1.4 Selección de lenguajes y herramientas para validar la solución 25
1.5 Trabajos relacionados 26
1.5 Conclusiones parciales 27

Capítulo 2 28
Ontología del modelo multidimensional 28
Introducción 28
2.1 Modelo Multidimensional y sus elementos 28
2.2 Elementos de la ontología del Modelo Multidimensional 29
Declaración de entidades en OWL2 32
Declaración de relaciones entre entidades en OWL2 36
2.4 Conclusiones parciales 41
Capítulo 3 42
Validación de la solución 42
Introducción 42
3.1 Propuestas para la evaluación de ontologías 42
3.2 Aplicación del método de evaluación a la propuesta 43
Fase 1: Uso correcto del lenguaje 44
Fase 2: Exactitud de la estructura taxonómica 45
Fase 3: Validez del vocabulario 46
Fase 4: Adecuación a los requerimientos 47
3.3 Aplicación de la solución a los datacubos del Censo y la Bolsa 48
Modelo multimensional Censo 49
Modelo multimensional Bolsa 54
3.3 Resultados y discusión 60
3.4 Conclusiones parciales 61
Conclusiones 63
Recomendaciones 65
Referencias Bibliográficas 66

Introducción

El procesamiento de datos en línea puede ser transaccional o analítico. El primero de ellos On-Line Transactional Processing, denominado OLTP por sus siglas en inglés, surge para lidiar con transacciones atómicas, cortas y aisladas, con un alto nivel de detalle y requiere que los datos estén al día. Las bases de datos que utilizan este tipo de procesamiento siguen un modelo lógico relacional. Todo esto es posible porque estas transacciones involucran un grupo relativamente pequeño de registros. Sin embargo este enfoque se torna ineficiente para realizar análisis, debido a que no estaba preparado para manejar datos consolidados, resumidos y de tipo histórico. Surge así el On-Line Analytical Processing (OLAP) donde los datos pueden cargarse de varias bases de datos desde las que se copia información de un largo período de tiempo. Por esta razón, las consultas tienen mayor complejidad, el tamaño de estos sistemas es notablemente superior y pasa a un segundo plano el procesamiento de transacciones para tomar su lugar el procesamiento de las consultas complejas y el tiempo de respuesta de las mismas.

Simultáneamente con OLAP surge una tecnología clave en las organizaciones, llamada almacenes de datos. Estos sistemas se utilizan como soporte a la toma de decisiones y también contribuyen a trazar los objetivos de la organización y su planeación estratégica. Tiene como finalidad resumir y centralizar información proveniente de diferentes fuentes para que sea de fácil acceso a los directivos de la empresa.

El diseño lógico de los almacenes de datos utiliza el modelo multidimensional que está compuesto por una tabla central llamada tabla de hechos relacionada con otras tablas que se llaman dimensiones. Esta representación en particular permite hacer análisis de información contenida en la tabla de hechos desde la perspectiva de las distintas dimensiones y con distintos niveles de detalle, llamado granularidad.

Con el paso del tiempo y el uso cada vez más generalizado de los almacenes de datos y las técnicas OLAP, la comunidad científica ha prestado especial interés a la integración de estas técnicas con la inteligencia artificial, y dentro de esta área con la minería de datos. Este proceso al igual que la minería tradicional, pretende encontrar material valioso a partir de una concentración del mismo. En este caso se trata de extraer conocimientos, en forma de patrones de una gran colección de datos para que sea utilizado por los directivos de la organización en el proceso de toma de decisiones. De esta manera, por ejemplo, determinar demandas máximas y mínimas, preferencias de los usuarios, temporadas de ventas y también predecir comportamientos de diferente índole, con el objetivo de establecer fortalezas organizacionales.

En el conocimiento extraído a través de la minería se utiliza algún tipo de lógica para proveer una semántica formal de cómo las funciones de razonamiento se aplican a las entidades del dominio. Entre las formas de representación existen las redes semánticas, los marcos, las reglas de producción y otras.

Las reglas de producción se representan de manera muy sencilla debido a que se puede deducir intuitivamente la relación entre los distintos elementos, por ejemplo, para la regla $\mathrm{A} \rightarrow \mathrm{B}$, si aparece A, entonces aparece B, por tanto es más fácil de comprender por el usuario.

La relación entre las reglas y el almacén está dada en que las primeras se extraen del segundo haciendo uso de un algoritmo de extracción. El más utilizado de todos es el A priori o alguna variante del mismo que tiene tres secciones. La primera de ellas está relacionada con la generación de candidatos a partir del conjunto original para esa iteración, la segunda es la poda de los candidatos que se conoce que no serán frecuentes, y la tercera relacionada con el conteo de las apariciones y se determina de acuerdo a un umbral si son frecuentes o no los itemsets[1]. Pero los algoritmos de extracción encuentran una cantidad tan grande de reglas debido al tamaño total de los datos contenidos en el almacén, que no pueden ser

[1] Conjunto de items

interpretadas por los especialistas humanos, lo cual limita la utilización de los modelos de reglas de asociación en la toma de decisiones pues se considera por el autor que el estudio de más de cien reglas de asociación por una persona pierde calidad y es de difícil procesamiento.

Existen un conjunto de métricas de calidad que definen la importancia de las reglas de asociación y contribuyen a establecer un orden para determinar cuáles son relevantes o no. Dos de estas métricas son el soporte y la confianza. La primera métrica de calidad es el soporte cuyo valor es el grado de pertenencia que tienen los elementos al conjunto difuso F sobre un universo de discurso o la proporción de transacciones en la base de datos que contiene dicho conjunto de ítems. La segunda métrica es la confianza cuyo valor es la probabilidad de encontrar la parte derecha de una regla condicionada a que se encuentre también la parte izquierda. (Molina 2005)

Una de las propuestas que puede mitigar este problema, es la integración del conocimiento previo del dominio dentro del proceso de generación de reglas. Los principales resultados los estudios se han alcanzado en variantes de minado con restricciones que utilizan las expectativas del usuario final para limitar el espacio de búsqueda a través de poda o generalización de elementos o limitar las reglas que forman parte del conjunto final, ya sea generalizando alguna de las reglas para que sean más comprensibles o eliminando reglas que no parezcan significativas al usuario.

Para reducir la complejidad de las reglas generadas por el algoritmo de extracción, presentadas al usuario final, es factible establecer relaciones a partir del conocimiento que posee un experto, que simplifiquen las reglas resultantes o limiten su cantidad. El uso de ontologías puede contribuir de manera directa a este propósito. Se trata en esencia de crear un lenguaje para representar las entidades y relaciones que existen en el almacén integrándolas con el conocimiento del experto. El mecanismo que sea entendido por una computadora donde los

conceptos se representen como entidades y sus relaciones puedan ser extendidas y compartidas, constituye la ontología. (Studer et al. 1998)

De forma aislada los almacenes de datos, la minería, las reglas de asociación, el conocimiento previo y las ontologías no contribuyen a mejorar la calidad del conocimiento extraído para presentarlo al usuario. Pero si se utiliza la minería de datos para extraer conocimiento del almacén en forma de reglas y se introducen elementos del conocimiento previo usando una ontología como mecanismo para agruparlos sí contribuye a que el resultado presentado al usuario final resulte más comprensible incrementando la calidad del producto.

Con motivo de esta situación y para guiar el desarrollo de la investigación se plantea lo siguiente:

Problema a resolver:
Los elementos para representar el conocimiento previo del negocio, en bases de datos multidimensionales, no propician el uso del mismo en la solución de problemas de minería de datos.

Objetivo General:
Desarrollar una ontología que permita representar de forma integrada la estructura de un cubo de datos y el conocimiento subyacente del dominio relacionado con las asociaciones de elementos.

Idea a defender:
Si se desarrolla una ontología que represente los elementos del modelo multidimensional y sus relaciones se obtienen modelos de dominio más efectivos para su utilización en la minería de reglas de asociación.

Objeto de Estudio:
Modelo multidimensional de datos

Campo de Acción:

Representación del modelo multidimensional de datos

Posibles resultados:

Ontología que represente el modelo multidimensional y las asociaciones de elementos en un cubo de datos

Objetivos Específicos:

1. Establecer el marco conceptual de referencia
2. Establecer un modelo teórico para la representación de asociaciones en el modelo multidimensional
3. Definir una ontología "core"[2] para la representación del modelo multidimensional
4. Definir un mecanismo para obtener la representación ontológica de un cubo de datos
5. Constatar la efectividad de la solución en un caso de estudio

Tareas a cumplir:

1. Recopilación de la bibliografía referente al tema
2. Selección de la bibliografía
3. Análisis de la bibliografía
4. Selección de los elementos esenciales que componen el modelo multidimensional y asociarle una representación ontológica
5. Selección de un mecanismo de representación ontológica del conocimiento previo sobre asociaciones de elementos en cubos de datos
6. Implementación de una aplicación que permita obtener la representación ontológica de un cubo de datos
7. Presentación de los resultados

[2] De propósito general o genérica

Descripción de los capítulos

En el Capítulo 1 se presentan los conceptos sobre los que se cimienta el trabajo de investigación realizado. Primeramente se exponen los conceptos fundamentales de almacenes de datos, modelo multidimensional y su estructura. Posteriormente se describen las reglas de asociación como resultado del proceso de minería de datos y el problema que representa su extracción. Finalmente se introducen elementos de conocimiento previo para analizar su efecto en la redundancia de las reglas de asociación y definir la forma en que se representa este conocimiento.

En Capítulo 2 se definen una ontología “core” y puntos de extensión para construir modelos de dominio a partir de información disponible en cubos de datos. Se toman los conceptos que se consideran relevantes del modelo multidimensional para representarlos como parte de la ontología y se desarrollan un grupo de ejemplos que permiten mostrar las transformaciones entre este modelo y los elementos de la ontología para representarlos en el lenguaje OWL2.

En Capítulo 3 se analizan las tendencias para la evaluación de ontologías en la actualidad y se hace un análisis para aplicarlo a la solución propuesta. Se decide usar un método que establece cuatro etapas para evaluar la solución y se realizan modificaciones para adaptarlo a este fin. Se muestran también dos ejemplos para mostrar el proceso de transformación.

Capítulo 1
Fundamentación Teórica

Introducción

En este capítulo se presentan los conceptos sobre los que se cimenta el trabajo de investigación realizado. Se abordan los conceptos fundamentales de almacenes de datos, haciendo énfasis en el modelo multidimensional y su estructura por su importancia para esta investigación. Se describen las reglas de asociación como resultado del proceso de minería de datos y se explica el problema que representa su extracción y la redundancia. Finalmente se introducen elementos de conocimiento previo para analizar su efecto en la redundancia de las reglas de asociación y definir la forma en que se representa este conocimiento.

1.1 Modelo multidimensional y OLAP

1.1.1 Almacenes de datos

Bill Inmon fue uno de los primeros autores en escribir sobre el tema de los almacenes de datos, define un almacén de datos en términos de las características del repositorio de datos es: una colección de datos orientados al tema, integrados, temporales, y no-volátiles para la toma de decisiones. (Inmon and Hackathorn 1994)

- Orientado a temas.- Los datos en la base de datos están organizados de manera que todos los elementos de datos relativos al mismo evento u objeto del mundo real queden unidos entre sí.
- Integrado.- La base de datos contiene los datos de todos los sistemas operacionales de la organización, y dichos datos deben ser consistentes.

- Variante en el tiempo.- Los cambios producidos en los datos a lo largo del tiempo quedan registrados para que los informes que se puedan generar reflejen esas variaciones.
- No volátil.- La información no se modifica ni se elimina, una vez almacenado un dato, éste se convierte en información de sólo lectura, y se mantiene para futuras consultas.

Ralph Kimball, otro conocido autor en el tema, define un almacén de datos como: "una copia de las transacciones de datos específicamente estructurada para la consulta y el análisis". (Kimball 1996)

Las definiciones de Inmon y Kimball se centran en los datos en sí mismos, pero para realizar la gestión de datos es importante conocer cómo extraerlos, transformarlos y cargarlos para su posterior análisis y producción de información útil para el usuario.

Esta herramienta integra datos de distintas fuentes, como son bases de datos, hojas de cálculo, archivos, entre otras, pero una diferencia con las bases de datos comunes es que los datos se encuentran integrados, como lo expresa el concepto antes referido. En el almacén se guardan copias del estado del sistema en momentos determinados constituyendo un registro histórico del comportamiento de la empresa. Solo se efectúan operaciones de consulta e inserción y no de eliminación ni de actualización, por lo tanto esta estructura deja de ser un conjunto para convertirse en una colección ya que pueden encontrarse varias repeticiones de los registros en el mismo momento.

Al disponer de este cúmulo de información, se pueden comparar períodos de ventas, inversiones, comportamiento del mercado, clientes y muchas otras variables que proveen a los decisores y directivos de una empresa los recursos necesarios para soportar y tomar sus decisiones. Igualmente esta gran cantidad de información es una poderosa herramienta que tiene la empresa y que puede utilizarse como una fuente de conocimiento.

El autor considera que se trata de una tecnología clave para complementar el proceso de toma de decisiones de las empresas en la actualidad, en lo cual radica su importancia y como también contribuyen a trazar los objetivos de la organización y su planeación estratégica, impacta en sus resultados.

1.1.2 OLAP y OLTP

On-Line Analytical Processing (OLAP) es una categoría tecnológica de software que permite a los analistas, directivos y ejecutivos acceder a los datos de forma rápida, consistente e interactiva a través de una amplia variedad de vistas de la información que han sido obtenidas de datos sin procesar para reflejar la dimensionalidad real de la empresa como la entiende el usuario. (OLAPCouncil)

OLAP surge para organizar grandes bases de datos y dar soporte a la inteligencia empresarial, con el objetivo de obtener datos completos y consistentes sobre el dominio a tratar.

Las operaciones de OLAP son:

- Roll-up para subir el nivel de granularidad en una jerarquía
- Drill-down para hacer lo contrario
- Slice se usa para restringir las dimensiones a analizar
- Dice para restringir valores de la tabla de hechos
- Pivot para hacer una tabulación cruzada

OLTP (On-Line Transactional Processing), surge para lidiar con transacciones atómicas, cortas y aisladas, con un alto nivel de detalle y requiere que los datos estén al día. Todo esto es posible porque estas transacciones involucran un grupo relativamente pequeño de registros. Sin embargo este enfoque se tornaba ineficiente para el procesamiento analítico de la información, debido a que no estaban preparados para esta finalidad, pues las exigencias son datos consolidados, resumidos y de tipo histórico.

El modelo lógico utilizado para representar estos datos es el relacional, en el que la base de datos se presenta en forma de relaciones, el cual es más comprensible si se formulan las relaciones como tablas que se conectan entre sí.

En OLTP a diferencia de OLAP, los datos consolidados pueden cargarse de varias bases de datos relacionales desde las que se copia información de un largo período de tiempo, razón por la cual las consultas son más complejas, el tamaño de estos sistemas es notablemente superior y pasa a un segundo plano el procesamiento de transacciones para tomar su lugar el procesamiento de las consultas complejas y el tiempo de respuesta de las mismas. Por lo tanto se hace necesario un nuevo modelo de datos para dar soporte a estos sistemas: el modelo multidimensional.

1.1.3 Modelo Multidimensional

El modelo de Agrawal fue una de las primeras formalizaciones que se ha hecho de un modelo multidimensional. Se propone un modelo con la siguiente funcionalidad:

- Tratamiento simétrico no solo de todas las dimensiones sino también de las medidas.
- Soporte para múltiples jerarquías en cada dimensión.
- Soporte para el cálculo de agregados ad-hoc, es decir, no se limitan los agregados a aquellos predefinidos.
- Soporte para un modelo de consulta. (Agrawal et al. 1995)

En un modelo de datos multidimensional hay un conjunto de hechos o medidas que son el objeto de análisis: ventas, presupuestos, inventario, entre otros. La mayoría de los hechos más útiles son numéricos, continuamente valuados y aditivos. La razón de que sean de dichas características es la siguiente: las consultas a la tabla de hechos necesitarán a su vez consultas de cientos, miles o incluso millones de registros para construir el conjunto respuesta. (Molina 2005)

El autor considera útil el modelo multidimensional como uno de los elementos para la representación de un almacén de datos.

En el modelo a utilizar "cada medida de análisis depende de un conjunto de Dimensiones, que proporcionan el contexto. En un espacio multidimensional para designar un punto son necesarias todas sus coordenadas, dependiendo del número de dimensiones que tenga el espacio. Esto es lo que ocurre en el modelo, ya que al indicar un valor para cada dimensión se situará un hecho concreto dentro del espacio formado." (Molina 2005)

Las definiciones de cada uno de estos elementos se presentan a continuación:

Dimensión

"Las dimensiones establecen el contexto del análisis sobre los hechos. Para acceder a los hechos a diferentes niveles de detalle, se pueden definir jerarquías sobre las dimensiones, estableciendo los niveles de granularidad posibles. Cada uno de estos niveles es un conjunto de nombres o etiquetas que definen subconjuntos de elementos de los niveles inferiores que agrupan." (Molina 2005)

Una dimensión sobre el dominio es una tupla $d = (l, \leq_d, l_\perp, l_\top)$ donde $l = \{l_1, \ldots, l_n\}$ tal que cada l_i es un conjunto $l_i = \{c_{i1}, \ldots, c_{im}\}$ tal que $l_i \cap l_j = \emptyset$ si $i \neq j, \leq_d$ es una relación de orden parcial tal que $l_i \leq_d l_k$ si $\forall c_{ij} \in l_i \Rightarrow \exists c_{kp} \in l_k / c_{ij} \subseteq c_{kp}$. $l_\perp$ y $l_\top$ son dos elementos de l tal que $\forall l_i \in l\ l_\perp \leq_d l_i$ y $l_i \leq_d l_\top$.

"Cada elemento li se denomina nivel. Para identificar el nivel li de la dimensión d se denota como $d.li$. Un elemento de la dimensión di y nivel lj se denota c_{jk}^{i} . Los niveles especiales $l\perp$ y $l\top$ se denominan nivel base y nivel superior respectivamente." (Molina 2005)

Hecho

"Las variables del dominio que se analiza, definen los hechos del DataCubo. Estos son los que en mayor medida acotan el dominio de análisis del DataCubo que se defina, limitando qué medidas podemos obtener. Las dimensiones aportan la contextualización de estas variables, localizando cada hecho dentro del espacio que definen. Cada uno de estos hechos es una agrupación de variables (una tupla)." (Molina 2005)

Considerando un conjunto de atributos $A_1, \ldots, A_n$ con dominios $D_1, \ldots, D_n$, se denomina hecho a cualquier $\mathrm{h} = (\mathrm{x}_1, \ldots, \mathrm{x}_n)$ tal que $\mathrm{x}_i \in D_i \; \forall_i = 1, \ldots, \mathrm{n}$, es decir, cualquier n-tupla definida sobre los dominios de los atributos que interesa estudiar. (Molina 2005)

DataCubo

Un DataCubo es una tupla $C = (D, l_b, F, H, A)$ tal que $D = (d_1, \ldots, d_n)$ es un conjunto de dimensiones, $l_b = (l_{1b}, \ldots, l_{nb})$ un conjunto de niveles tal que l_{1b} pertenece a d_i, $F = R \cup \emptyset$ donde R es el conjunto de hechos y $\emptyset$ un símbolo especial, H es un objeto de tipo historia, y A es una aplicación definida como $A: l_{1b} \times \ldots \times l_{nb} \rightarrow F$, que para cada conjunto de valores de las dimensiones devuelve el hecho relacionado con estas coordenadas. (Molina 2005)

"Un DataCubo está formado por las estructuras antes definidas. Así pues, tiene una componente que representa los hechos a analizar. Otro de los elementos que se utilizan al definir un DataCubo son las dimensiones que caracterizan a estos hechos. En este caso son un conjunto ordenado de dimensiones. Además, se tiene en cuenta a qué nivel de cada dimensión se definen los hechos y una aplicación que diga para cada coordenada que definen los valores de estos niveles, el hecho relacionado." (Molina 2005). A consideración del autor dimensión, hecho y DataCubo son conceptos relevantes dentro del modelo multidimensional y que contribuyen a la representación del almacén de datos.

Para ilustrar la relación entre los conceptos anteriormente mencionados se muestra la Figura 1.1 para facilitar su comprensión.

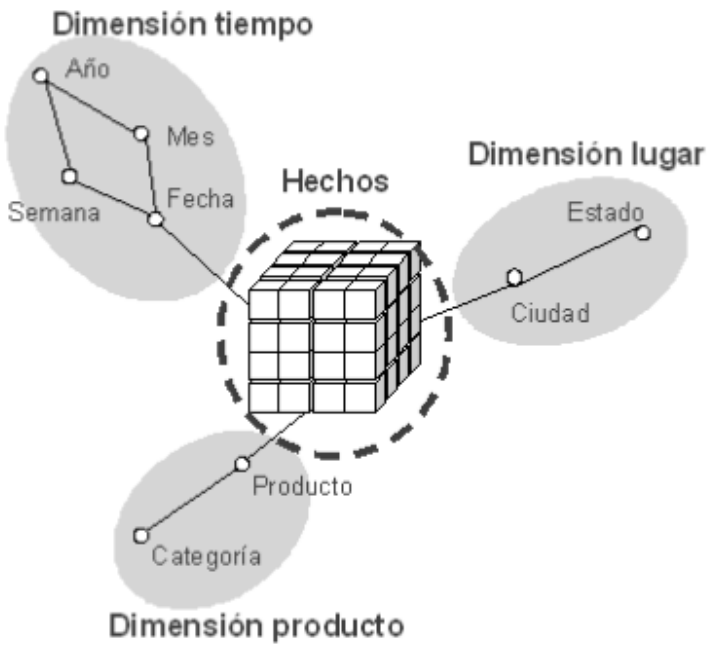

Figura 1.1: Estructura del modelo multidimensional

1.2 Minería de datos

1.2.1 Knowledge Discovery in Databases

La minería de datos se define como “el descubrimiento automático por métodos de machine-learning de patrones previamente desconocidos o relaciones en largos y complejos conjuntos de datos (…bases de datos, almacenes de datos)” (Rennolls 2005)

Es un proceso que intenta descubrir patrones en conjuntos de grandes volúmenes de datos a través de los métodos de la inteligencia artificial, aprendizaje automático, estadística y sistemas de bases de datos, consiste en extraer información, transformarla y hacerla más comprensible para su uso posterior.

La minería de datos es una fase del Knowledge Discovery in Databases (KDD) a la cual antecede la Recopilación e integración y la Selección, limpieza y transformación, también conocido como pre-procesamiento. En estas fases se

extraen los datos de distintas fuentes para disponerlos de forma unificada y accesible y se tratan para eliminar inconsistencias, datos repetidos, datos anómalos, rellenar valores y suavizar el ruido. La minería, encargada de extraer conocimiento en forma de patrones es complementada posteriormente con la evaluación, difusión y uso. (DAEDALUS 2002), (Martínez 2003) y (Vallejos 2006). Esta estructura se muestra en la Figura 1.2

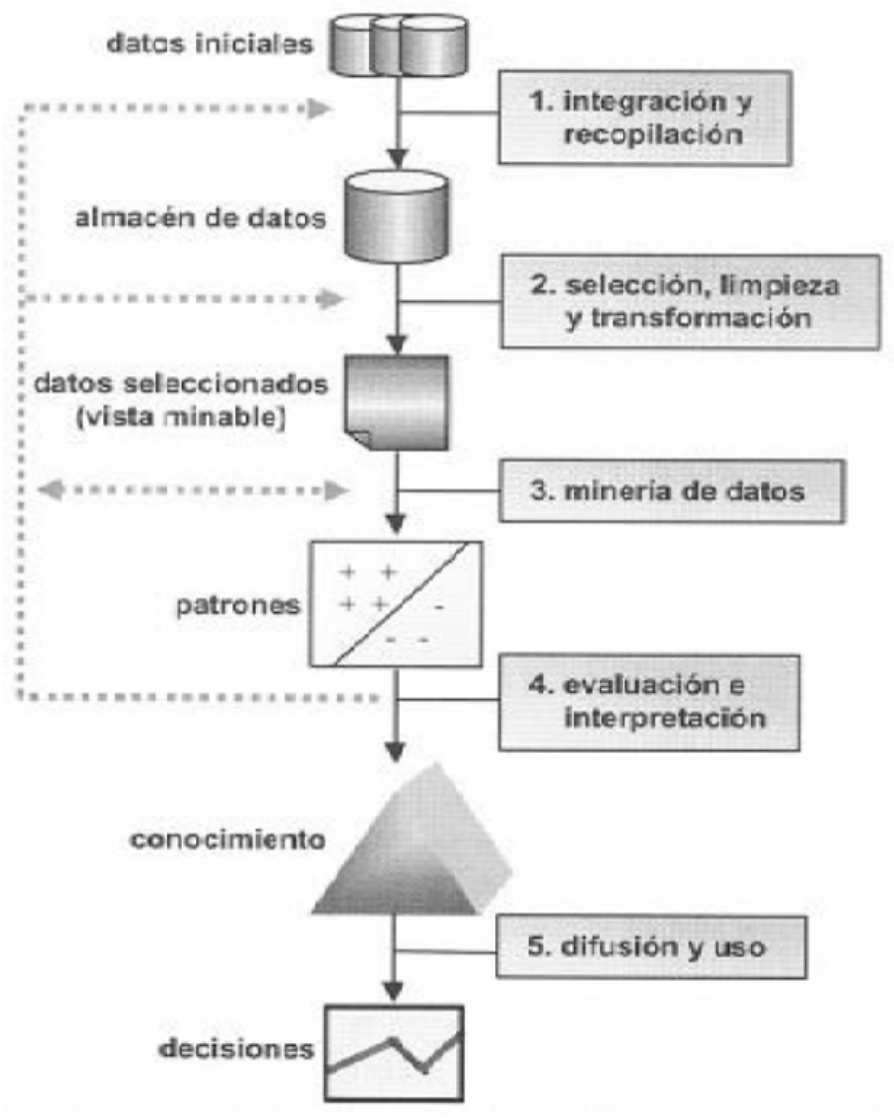

Figura 1.2: Fases de KDD (Knowledge Discovery in Databases)

La entrada del proceso de KDD es el almacén de datos del cual se extrae conocimiento valioso y su salida, conocimiento escrito en forma de reglas de asociación para que sea usado en el proceso de toma de decisiones.

1.2.2 Reglas de asociación

Reglas de asociación son las relaciones entre variables que se utilizan para descubrir hechos que ocurren en común dentro de una determinada colección de datos, en el caso de esta investigación, el DataCubo. Este proceso se lleva a cabo como parte de la minería de datos la cual busca extraer información no explícita sobre grandes volúmenes de datos.

Normalmente se buscan relaciones entre las tuplas o entre los atributos, estas asociaciones se pueden representar de diversas formas. Entre las formas de representación se encuentran: dependencias funcionales, reglas de asociación, dependencias funcionales aproximadas, dependencias funcionales difusas, reglas de asociación difusas, datos inconsistentes, patrones, anomalías, entre otras. El autor utiliza para la representación del conocimiento extraído del DataCubo, las reglas de asociación. (Diaz et al. 2013)

El autor Agrawal define regla de asociación como:

"Sea $I = \{ i_1, i_2, \ldots, i_2 \}$ un conjunto de literales y D un conjunto de transacciones definidas sobre I. Una regla de asociación es una implicación de la forma $X \rightarrow Y$, donde $X \subset I, Y \subset I$ y $X \cap Y = \emptyset$." (Agrawal et al. 1993)

1.2.3 Soporte y confianza

Las reglas de asociación tienen dos medidas que se utilizan para determinar su calidad, ellas son el soporte y la confianza. El primero Molina lo define como:

El soporte de un conjunto difuso F definido sobre U es el conjunto de elementos con pertenencia mayor que 0 (Soporte (F) = $\{u \in U/\mu F(u) > 0\}$), es decir, todos aquellos elementos que pertenecen en algún grado al conjunto. (Molina 2005). Siendo U el universo de discurso.

El segundo término el mismo autor lo define como:

Una regla $X \rightarrow Y$ tendrá en D una confianza con valor c si el $c\%$ de las transacciones en D que contienen X también contienen Y. La regla tendrá un soporte con valor s en el conjunto de transacciones D si el $s\%$ de las transacciones en D contienen a los elementos $X \cup Y$.

El valor de la primera es la proporción de transacciones en la base de datos que contiene dicho conjunto de ítems, expresado en la Fórmula 1:

$$sop(X) = \frac{|X|}{|D|} \qquad [1]$$

Y la segunda es la probabilidad de encontrar la parte derecha de una regla condicionada a que se encuentre también la parte izquierda cuya definición matemática está en la Fórmula 2:

$$\mathrm{conf}(X \Rightarrow Y) = \frac{sop(X \cup Y)}{sop(X)} \qquad [2]$$

1.2.4 Extracción de reglas y Algoritmo de extracción

Una de las técnicas más utilizadas en la minería de datos es la extracción de reglas de asociación que se descompone en dos pasos:

1. Descubrir los conjuntos de elementos frecuentes, es decir, los conjuntos de elementos con soporte superior a un umbral prefijado.
2. Utilizar los conjuntos de elementos frecuentes para generar reglas de asociación.

El primero de los pasos es el que consume mayores recursos de tiempo y espacio. La generación de reglas, una vez que se tienen los conjuntos de elementos frecuentes, es simple. Con el fin de obtener estos conjuntos se han propuesto múltiples algoritmos, tomando como punto de partida uno en particular y después intentando mejorar su eficiencia. Ejemplos de estos son el algoritmo Apriori, AprioriTID y AprioriHybrid, DHP (el cual utiliza tablas hash para acelerar la localización de candidatos a conjuntos frecuentes, Eclat (el cual calcula el soporte realizando intersecciones de conjuntos que aborta en cuanto puede asegurar que no se alcanzará el soporte mínimo. (Molina 2005)

A pesar de que existen mejoras del algoritmo Apriori, es muy utilizado en la extracción de reglas de asociación y por tanto el conocimiento que se obtiene después de aplicar este algoritmo sobre un DataCubo, es el que se usará en la presente investigación. Por su importancia teórica, se describen algunos aspectos sobre este algoritmo a continuación.

Apriori está basado en el principio de “clausura descendente del soporte de ítemsets”, la cual plantea:

Clausura descendente del soporte: El soporte de cualquier subconjunto de un conjunto de ítems es mayor o igual que el soporte de ese conjunto de ítems. Lo que implica que todo subconjunto de un conjunto frecuente de ítems es también frecuente.

Un ítemset[3] es frecuente cuando cumple con un soporte mínimo definido con anterioridad. Es importante destacar que Apriori asume que los ítems de cualquier transacción están ordenados lexicográficamente, para una correcta generación de los k-ítem sets candidatos. (Motoda and Ohara 2009)

A priori está compuesto por tres secciones para la obtención de los k-ítemsets frecuentes:

- Generación de candidatos: se generan los candidatos de nivel k (k-1 ítemsets) haciendo uso de la información referida a los conjuntos frecuentes de la iteración anterior (k-1 ítemsets frecuentes).
- Poda: se garantiza que no pase al conteo de soporte los ítemsets que se conoce a priori que no van a ser frecuentes, eliminando a todos aquellos candidatos que no cumplan con que todos sus subconjuntos sean también frecuentes.
- Conteo de soporte: se cuenta la cantidad de apariciones de los k-ítemsets candidatos en la base de datos y se determina, según el soporte mínimo establecido por el usuario, el conjunto de los k-ítemsets frecuentes.

Una vez generados todos los ítemsets frecuentes se procede a generar todas las posibles reglas de asociación por cada uno de ellos. Para cada k-ítemset frecuente se puede generar $2^{\mathrm{k}} - 2$ reglas de asociación, ignorando aquellas reglas que carezcan de antecedente o consecuente $(\{\emptyset \rightarrow A\}, \{A \rightarrow \emptyset\})$. Las reglas se pueden extraer dividiendo el ítemset en dos subconjuntos no vacíos, por ejemplo, el ítemset $\{A, B\}$ puede dividirse en los subconjuntos $\{A\}$ y $\{B\}$, de los cuales pueden obtenerse las reglas $\{A \rightarrow B\}$ y $\{B \rightarrow A\}$, estas deben cumplir con los umbrales de soporte y confianza definidos por el usuario. (Agrawal and Sritkant 1994)

En los almacenes de datos y modelos multidimensionales en general, el problema de la extracción de reglas de asociación (Alhajj and Kaya 2003), (Zhu 1998) se dificulta ya que se disponen de datos en varias dimensiones y se pueden obtener

[3] Conjunto de items

un mayor número de reglas. El post-procesamiento de las reglas para eliminar reglas redundantes o para buscar las más significativas también aumenta de dificultad. Por tanto, se presenta un problema de gran complejidad pero, a la vez, de gran importancia y utilidad. (Molina 2005)

Otro elemento a tener en cuenta es que los datos pueden presentar jerarquías, las cuales pueden surgir mediante la combinación de atributos en las dimensiones como el caso de las fechas agrupándolas por meses, años, semestres, trimestres, entre otros o dadas de forma explícita por expertos del proceso. Las relaciones entre los elementos es posible que no se encuentren en los niveles más bajos de las jerarquías y sí en niveles más altos. (Diaz, Molina and Vila 2013)

El resultado de ejecutar un algoritmo de extracción es el conocimiento extraído del DataCubo, expresado en reglas de asociación. El autor considera para esta investigación, incluir este conocimiento junto con la estructura del modelo multidimensional para simplificar el conjunto de reglas mostrado al usuario.

1.2.5 Ejemplo de reglas generadas

Los algoritmos de extracción de reglas de asociación suelen generar un conjunto muy grande de reglas que resulta intratable por el usuario. Este inconveniente ha provocado que se encaminen esfuerzos a desarrollar paralelamente al proceso, métodos para la reducción del número de reglas obtenido o su complejidad.

En el ejemplo siguiente se utiliza un algoritmo de extracción que tiene en cuenta el conocimiento aportado por un experto y por lo tanto es necesario definir el concepto de redundancia del conocimiento previo.

Sea S un conjunto de reglas de asociación y S_c un conjunto de reglas conocidas con confianza igual a 1, definidas sobre el mismo dominio de S. Una regla de asociación $R: X \rightarrow Y \in S$ es redundante con respecto a S_c si hay una regla $R': X' \rightarrow Y' \in S_c$ y satisface alguna de las condiciones siguientes (Diaz et al. 2014):

1. $X' \subseteq X \wedge Y' \subseteq Y$
2. $X' \subseteq X \wedge \exists R'': X'' \rightarrow Y'' \epsilon\, S_c: Y \subseteq Y'$
3. $Y \subseteq Y' \wedge X' \subseteq X$
4. $X' \subseteq X \wedge Y' \subseteq X$
5. $X' \subseteq Y \wedge Y' \subseteq Y$

Después de esta definición se muestra un ejemplo de una base de datos a la que se le aplica el algoritmo Apriori y se obtienen las reglas de asociación.

Tabla 1.1 Datos iniciales para algoritmo Apriori

oid	items	soporte	Itemsets
1	A,C,D	1/5	D
2	B,C,E	2/5	AB, AE, ABC, ABE, ACE, ABCE
3	A,B,C,E	3/5	A, AC, BC, CE, BCE
4	B,E	4/5	B, C, E, BE
5	A,B,C,E	1	$\{\emptyset\}$

En la parte izquierda de la tabla anterior se muestran las transacciones de una base de datos y a la derecha se muestra el soporte y los itemset. Las reglas de asociación $B \rightarrow C$ y $C \rightarrow E$ se tienen como conocimiento previo. Para ejecutar el algoritmo se usa un umbral para el soporte y la confianza del 40%. El conjunto de reglas fuertes es: $\{A \rightarrow B, B \rightarrow A, A \rightarrow E, E \rightarrow A, A \rightarrow BC, AB \rightarrow C, AC \rightarrow B, B \rightarrow CA, BC \rightarrow A, C \rightarrow AB, A \rightarrow BE, AB \rightarrow E, E \rightarrow AB, A \rightarrow CE, AC \rightarrow E, AE \rightarrow C, C \rightarrow AE, CE \rightarrow A, E \rightarrow AC, A \rightarrow BCE, AB \rightarrow CE, AC \rightarrow BE, AE \rightarrow BC, ABC \rightarrow E, ABE \rightarrow C, ACE \rightarrow B, B \rightarrow ACE, BC \rightarrow AE, BE \rightarrow AC, BCE \rightarrow A, C \rightarrow ABE, CE \rightarrow AB, E \rightarrow ABC, A \rightarrow C, C \rightarrow A, B \rightarrow C, C \rightarrow B, C \rightarrow E, E \rightarrow C, B \rightarrow CE, BC \rightarrow E, BE \rightarrow C, C \rightarrow BE, CE \rightarrow B, E \rightarrow BC, B \rightarrow E, E \rightarrow B\}$.

La regla $A \rightarrow BC$ satisface la condición 5 de la definición de redundancia con respecto a la regla $B \rightarrow C$ de S_c $(B \subseteq B \wedge C \subset BC)$ por lo tanto la regla es redundante con respecto al conocimiento previo S_c. Ahora tomando la regla $AB \rightarrow C$ y $B \rightarrow C$ de S_c se tiene que $(C \subseteq C \wedge B \subset BC)$ por lo que la regla satisface la condición 3 de la

definición y es redundante. La regla $B \rightarrow CA$ es redundante con respecto a $B \rightarrow C$ de S_c porque satisface la condición 1 en la definición de redundancia ($B \subset B \wedge C \subseteq CA$). La regla de asociación $BC \rightarrow A$ cumple con la condición 4 de la definición de redundancia con respecto a la regla $B \rightarrow C$ ($B \subseteq B \wedge C \subseteq C$). La regla $AB \rightarrow E$ satisface la condición 2 de la definición de redundancia tomando $B \rightarrow C$ como R' y $C \rightarrow E$ como R'' ($B \subset AB \wedge C \subseteq CA$).

Se muestra a continuación, la lista completa de reglas redundantes en la siguiente tabla, por cada una de las condiciones con las que coincide.

Tabla 1.2 Reglas redundantes descubiertas por el algoritmo Apriori

	1	2	3	4	5
1	$B \rightarrow CA$	$AB \rightarrow E$	$AB \rightarrow C$	$BC \rightarrow A$	$A \rightarrow BC$
2	$C \rightarrow AE$	$A \rightarrow BE$	$AC \rightarrow E$	$CE \rightarrow A$	$A \rightarrow CE$
3	$B \rightarrow ACE$	$B \rightarrow AE$	$AB \rightarrow CE$	$ACE \rightarrow B$	$A \rightarrow BCE$
4	$BC \rightarrow AE$	$AC \rightarrow BE$	$ABC \rightarrow E$	$BCE \rightarrow A$	$AE \rightarrow BC$
5	$BE \rightarrow AC$	$C \rightarrow ABE$	$ABE \rightarrow C$	$CE \rightarrow AB$	$E \rightarrow ABC$
6	$B \rightarrow CE$	$B \rightarrow E$		$CE \rightarrow B$	$BC \rightarrow E$
7	$BE \rightarrow C$				$E \rightarrow BC$
8	$C \rightarrow BE$				

1.3 Ontologías

1.3.1 Conocimiento previo

Un enfoque que se emplea para reducir la cantidad y complejidad del conjunto de reglas obtenido por el algoritmo de extracción, es usar conocimiento del dominio que posee un experto. La ventaja de hacerlo es que ninguna máquina puede acumular las experiencias que posee una persona, generadas en su trabajo con el surgimiento de situaciones en el negocio. Escribir estas experiencias en forma de reglas de asociación, permite reducir las reglas obtenidas por el algoritmo de

extracción ya sea podándolas o generalizándolas. Esto se realiza con el objetivo de que el conjunto de reglas que se muestre al usuario final, sea lo más reducido posible y no redundante, pues para un ser humano es complicado el análisis de más de cien reglas de asociación.

Con el fin de integrar toda esta información junto con los metadatos del almacén es necesario utilizar alguna estructura o artefacto (se usa este término porque debe ser entendido por una computadora) donde están contenidas un conjunto de relaciones que expresan las asociaciones entre las distintas clases del dominio a tratar. (Diaz, Molina and Vila 2013; Diaz, Molina and Vila 2014)

Existen muchas formas de representar estas relaciones y se conocen formalmente como formas de representar el conocimiento. De manera general se destacan las redes semánticas, los marcos o frames, los guiones o scripts, las reglas de producción y otras igualmente importantes. Con el desarrollo de lo que se conoce como la Web Semántica ha surgido una nueva forma de representar conocimiento aplicado a la informática, en este caso se hace referencia al término ontología.

1.3.2 Definición de ontología

Según Guarino, una ontología se puede considerar como un conjunto de axiomas lógicos que se han diseñado para considerar el significado propuesto de un vocabulario. Dado un lenguaje L con compromiso ontológico K, una ontología para L es un conjunto de axiomas diseñados en una manera tal que el conjunto de sus modelos aproxima lo mejor posible el conjunto de modelos esperado de L de acuerdo a K. (Guarino 1998)

Una ontología según la W3C generalmente se describe como un artefacto que contiene un conjunto de enunciados precisos descriptivos acerca de una parte del mundo. (W3C_Recommendation)

La definición adoptada por el autor y con la que se trabajará en lo adelante es la de Studer que expresa “una ontología es una especificación formal y explícita de una conceptualización compartida”. (Studer, Benjamins and Fensel 1998)

Se elige esta definición porque expresa de forma más clara lo que es una ontología. Por conceptualización se refiere a un modelo abstracto de algún fenómeno en el mundo identificando sus conceptos relevantes. Especificación explícita significa que el tipo de conceptos y las restricciones usadas están explícitamente definidas. Formal alude a que debe ser entendido por la computadora. Compartida sugiere que la ontología captura conocimiento que no es privativo para algunos individuos sino aceptado por un grupo. (A.Martin et al. 2011)

Las entidades y relaciones que se representan corresponden con las del modelo multidimensional y se usan extensiones de este modelo para describir el dominio de datos, aportado por el almacén y que sea entendido por una computadora.

Una ontología cuenta con tres elementos principales para representar el conocimiento. El primero de ellos son los axiomas que expresan premisas que se cumplen para todos elementos de la ontología y además siempre son verdaderos, esto es lo que los distingue de los restantes elementos. Por ejemplo si existiese una entidad, término que se define más adelante, llamada persona y otra llamado humano, un axioma constituiría que toda persona es un humano. El segundo elemento que compone la ontología son las entidades y se utilizan para representar elementos del mundo real que se quieran registrar. Retomando el ejemplo anterior, la entidad persona representa un elemento del mundo real, también puede haber jerarquías entre estas entidades y las entidades pueden tener individuos que se utilizan para explicitar instancias de las entidades. El último de los elementos son las expresiones que definen combinaciones de entidades para formar descripciones desde complejas hasta sencillas. Por ejemplo si se tiene la entidad Persona y la entidad María una expresión válida es que María es una persona o si se tiene la entidad María y la entidad Pedro, otra expresión válida sería que Pedro es esposo de María, es necesario tener en cuenta que estas relaciones no funcionan en los

dos sentidos, al igual que en la realidad, es incorrecto decir que María es esposo de Pedro. Lograr que las expresiones sean reflexivas implicaría la declaración de nuevas expresiones que lo expliciten. (W3C_Recommendation) (Lapuente 2013)

Según su dependencia y relación con una tarea específica, Guarino clasifica las ontologías en (Guarino et al. 2009):

- Ontologías “core”, de Alto nivel o Genéricas describen conceptos más generales.
- Ontologías de Dominio describen un vocabulario relacionado con un dominio genérico.
- Ontologías de Tareas o Técnicas básicas describen una tarea, actividad o artefacto, por ejemplo componentes, procesos o funciones.
- Ontologías de Aplicación describen conceptos que dependen tanto de un dominio específico como de una tarea específica, y generalmente son una especialización de ambas.

El autor utiliza una ontología “core” y una ontología de dominio porque la primera de ellas, como es más general, permite modelar los conceptos del modelo multidimensional y se utiliza la segunda para instanciar estos conceptos como un dominio a través de los datos contenidos en el almacén.

1.3.3 Lenguajes para representar ontologías

Las ontologías se pueden representar con varios lenguajes, por ejemplo, RDF, OWL y otros. Utilizándose este último porque brinda una mejor comprensión tanto a las personas como a la computadora. También OWL usa una sintaxis similar a XML para representar clases, individuos y relaciones entre ellos (Horrocks et al. 2003).

La W3C define el lenguaje OWL y una versión más avanzada OWL2 como:

“Un lenguaje web semántico diseñado para representar conocimiento rico y explícito sobre clases, grupos de clases y relaciones entre ellas. OWL es un lenguaje computacional basado en la lógica tal que el conocimiento expresado en OWL puede ser razonado por programas de computadoras, tanto para verificar la consistencia de ese conocimiento o para transformar el conocimiento implícito en explícito.” (W3C_Recommendation)

El autor considera usar este lenguaje en la presente investigación para declarar cada uno de los elementos de la ontología por sus ventajas y similitudes con los lenguajes marcados como XML y su flexibilidad para ser entendido por un software de computadora.

1.4 Selección de lenguajes y herramientas para validar la solución

Java es un lenguaje de programación de propósito general, concurrente, orientado a objetos y basado en clases. Es un lenguaje interpretado por lo tanto, el código es ejecutado en una máquina virtual lo que lo hace liviano y permite que se utilice en cualquier plataforma. Java es, uno de los lenguajes de programación más populares, particularmente para aplicaciones de cliente-servidor de web. (Gosling et al. 2005). La selección de este lenguaje se basa en que para la validación de la solución se utiliza el software FCubeFactory, desarrollado en el lenguaje Java. Para mantener la coherencia con la herramienta, se implementa el código de las transformaciones en este lenguaje.

NetBeans es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extenderlo y ofrece ventajas para la corrección de errores. Todo esto justifica su utilización. NetBeans IDE es un producto libre y gratuito sin restricciones de uso, por las razones antes mencionadas ambos se utilizan en esta investigación. (Domínguez-Dorado 2005).

1.5 Trabajos relacionados

El uso de ontologías aplicadas a los almacenes de datos y a la minería no es un tema nuevo. En realidad se han realizado trabajos con diferentes enfoques para utilizarlas con el fin de mejorar el rendimiento, simplificar las estructuras, obtener información, entre otros. Se ha hecho una revisión de algunos artículos publicados en los últimos diez años que tratan del uso de las ontologías para este fin con muy buenos resultados (Rennolls 2005), pero con objetivos diferentes al que se traza esta investigación. En este caso se usa este artefacto para lograr una mejor coherencia entre los procesos de minería de datos, descubrimiento de conocimiento (aprendizaje automático) y almacenes de datos. Otra publicación también hace referencia a resolver problemas (Wu et al. 2007) que se presentan en almacenes de datos, extracción de reglas de asociación en estos y la propuesta de un marco de trabajo para dichas operaciones. De igual forma, este enfoque no es el que se sigue en esta investigación ya que no están directamente relacionadas con la misma pero si aportan algunos elementos conceptuales que se tienen en cuenta en este capítulo, específicamente algunas experiencias en la extracción de reglas de asociación. Otra autora que ha publicado con relación a este tema es Cláudia Antunes, proponiendo un marco de trabajo que usa una ontología de dominio (Antunes 2009) para extraer patrones en almacenes de datos que sigan un modelo multidimensional con esquema de estrella y con posibilidad de manejar jerarquías en sus dimensiones, este enfoque se acerca al de esta investigación, pero la diferencia radica en que aquí se usó una ontología del dominio en lugar de representar el dominio como extensión de los elementos del almacén de datos. Es importante mencionar que el enfoque de esta autora es el que se menciona en la mayoría de los artículos para resolver problemas de este tipo. También existen otros trabajos que presentan esfuerzos para mejorar la calidad en el diseño de los almacenes de datos (Zheng et al. 2010) de acuerdo a los requerimientos del usuario, usando ontologías, igualmente este enfoque no tributa directamente a lo realizado en esta investigación pues no se representan dominios de datos.

En 2011 se han presentado resultados en usar ontologías (Xu et al. 2011) para transformar modelos relacionales a modelos multidimensionales. Por otra parte los trabajos de Nebot utilizan información contenida en la web 2.0 en forma de anotaciones (Nebot et al. 2011) para enriquecer el conocimiento del dominio que se tiene. Martin adiciona información de noticias disponibles en la web (Martin et al. 2011) para complementar el proceso de toma de decisiones. En los últimos tres casos se usan también ontologías, pero con el objetivo de mejorar directamente el diseño del almacén y la información contenida en él, a diferencia del enfoque que se sigue en esta investigación que se centra en crear una ontología para representar un dominio de datos junto con el conocimiento de un experto para hacer simplificar la forma y reducir la cantidad de conocimiento extraído por minería de datos que se presenta al usuario final.

1.5 Conclusiones parciales

El marco conceptual de referencia para soportar los fundamentos teóricos de la investigación permitió definir los almacenes de datos y sus elementos como parte del modelo multidimensional. Se introduce también, la minería de datos como proceso para extraer conocimiento del almacén en forma de reglas que también se usan para expresar el conocimiento previo. Todo lo anterior sustenta la propuesta de creación de la ontología que integra la estructura multidimensional del almacén con el conocimiento previo que es la propuesta de esta investigación.

Capítulo 2

Ontología del modelo multidimensional

Introducción

En el presente capítulo se definen una ontología “core” y puntos de extensión para construir modelos de dominio a partir de información disponible en cubos de datos. Se examinan los conceptos que se consideran relevantes del modelo multidimensional para representarlos como parte de la ontología, y también se desarrollan un grupo de ejemplos que permiten sistematizar las transformaciones entre este modelo y los elementos de la ontología de forma que puedan ser representados en el lenguaje OWL2.

2.1 Modelo Multidimensional y sus elementos

En el capítulo anterior se presentó la definición de modelo multidimensional de Molina, según este autor:

Un DataCubo es una tupla $C = (D, l_b, F, H, A)$ tal que $D = (d_1, \dots, d_n)$ es un conjunto de dimensiones, $l_b = (l_{1b}, \dots, l_{nb})$ un conjunto de niveles tal que l_{1b} pertenece a d_i, $F = R \cup \emptyset$ donde R es el conjunto de hechos y $\emptyset$ un símbolo especial, H es un objeto de tipo historia, y A es una aplicación definida como $A: l_{1b} \times \dots \times l_{nb} \rightarrow F$, que para cada conjunto de valores de las dimensiones devuelve el hecho relacionado con estas coordenadas. (Molina 2005)

La ontología que se propone en este capítulo incluye una representación para cada uno de los cinco elementos establecidos por Molina, exceptuando el objeto de tipo historia que se refiere a las transformaciones por las que ha pasado el almacén y

por lo tanto no se considera relevante porque el proceso de minería de datos se realiza sobre los datos que contiene el almacén y no sobre sus transformaciones. Además se agrega un tipo de relación que no corresponde con ninguno de los anteriores elementos, pero se usa para representar el conocimiento que posee un experto del negocio, el cual se considera relevante y de esta manera enriquecer la ontología. Con estos elementos se crea una ontología “core” o de propósito general para expresar los conceptos y relaciones contenidos en el modelo multidimensional y modelar la información perteneciente a un dominio, dada por el tema a que esté orientado el cubo de datos, con una ontología de dominio a través de puntos de extensión. De esta forma se pueden representar los conceptos y relaciones presentes en un dominio específico describiéndolos como especializaciones de los elementos del modelo multidimensional.

Se aclara que para la ontología genérica se utilizan, el DataCubo para identificar el dominio a describir, cada dimensión con cada atributo y los valores que toma cada uno. También es necesario establecer relaciones entre las clases de la ontología:

1. El DataCubo y cada una de las dimensiones
2. Cada dimensión y cada elemento de dimensión
3. Entre elementos de dimensión (aportada por el experto del dominio)

2.2 Elementos de la ontología del Modelo Multidimensional

En este documento se describe una Ontología del modelo multidimensional que está compuesta por una ontología “core” para representar cubos de datos y una de dominio para representar este a través de puntos de extensión de la primera. Se define esta estructura formal basada en los conceptos: Cubo, Dimensión, Elemento de dimensión, Hecho, Medida y Operador de agregación. Todos estos elementos agrupados en la Ontología del modelo multidimensional permiten crear un marco de trabajo estructural para representar y organizar ontologías de dominio específico a partir de estos puntos de extensión.

Para describir las relaciones entre los elementos que se presentan en la ontología se muestra la Figura 2.1:

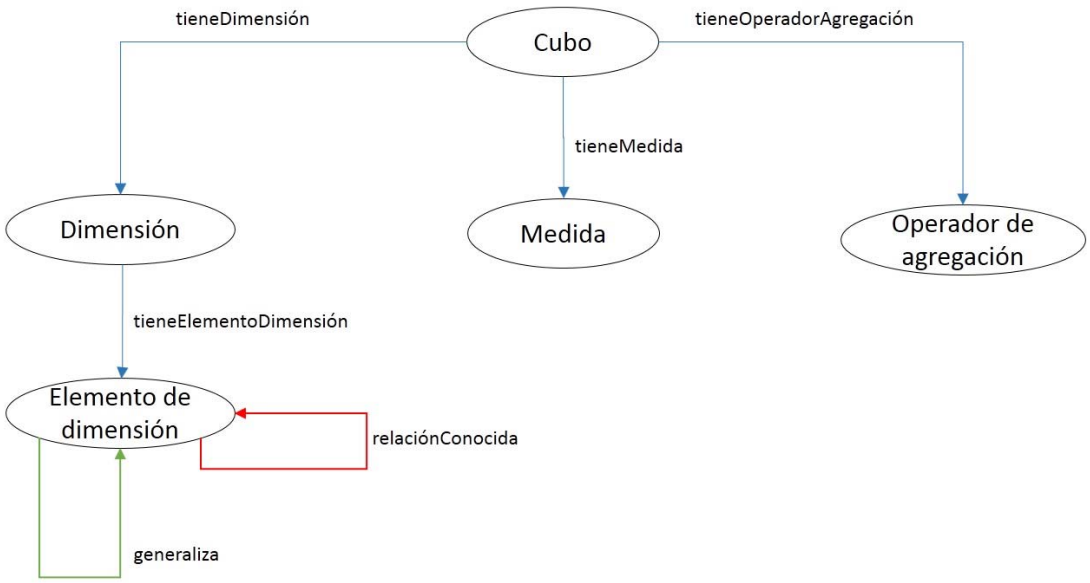

Figura 2.3: Relación entre los elementos de la ontología "core" del modelo multidimensional

En la Figura 2.1 se declaran los conceptos correspondientes a la ontología de propósito general o "core" que contiene los elementos genéricos pertenecientes al cubo de datos. El concepto Cubo es la entidad padre y contiene a las demás. En el segundo nivel se tiene en la izquierda a Dimensión unida a Cubo a través de la propiedad "tieneDimensión". En el nivel inferior se encuentra el concepto ElementoDimensión unido a Dimensión por la propiedad "tieneElementoDimensión". Entre los elementos de dimensión pueden existir también dos relaciones, una de ellas es la propiedad "generaliza" que se utiliza para reflejar las relaciones jerárquicas entre los elementos de una dimensión. La otra asociación que se puede presentar es "tieneRelaciónConocida" que no tiene correspondiente en el modelo multidimensional, pero es la que se emplea para expresar el conocimiento que posee el experto del dominio y de esta forma enriquecer la ontología. En el centro y derecha se encuentran los conceptos Medida y OperadorAgregación, ambos unidos a Cubo a través de la relación "tieneMedida" y "tieneOperadorAgregación" respectivamente. La información del dominio se

obtiene a partir de introducir los conceptos del dominio a tratar en la ontología como extensiones de los conceptos del almacén de datos, o sea, la información del dominio se expresa a través de especificaciones de las entidades y relaciones del almacén.

En la Figura 2.3 se ilustra una ontología de dominio siguiendo este proceder para representar información de un almacén de datos hipotético que cuenta con un DataCubo para realizar el análisis de las ventas en una determinada tienda, en lo adelante VentasEncanto. Los hechos presentados son las ventas de la tienda con dos medidas, la cantidad y el monto de las mismas. Las dimensiones para analizar dichas ventas son el Producto del que se conoce su Nombre, el Cliente que se puede agrupar por Sexo o por su Edad y por último la Fecha con posibilidad de jerarquizar por el Mes o Año de la compra. Para este almacén se tiene, siguiendo la topología de estrella, el modelo de la Figura 2.2.

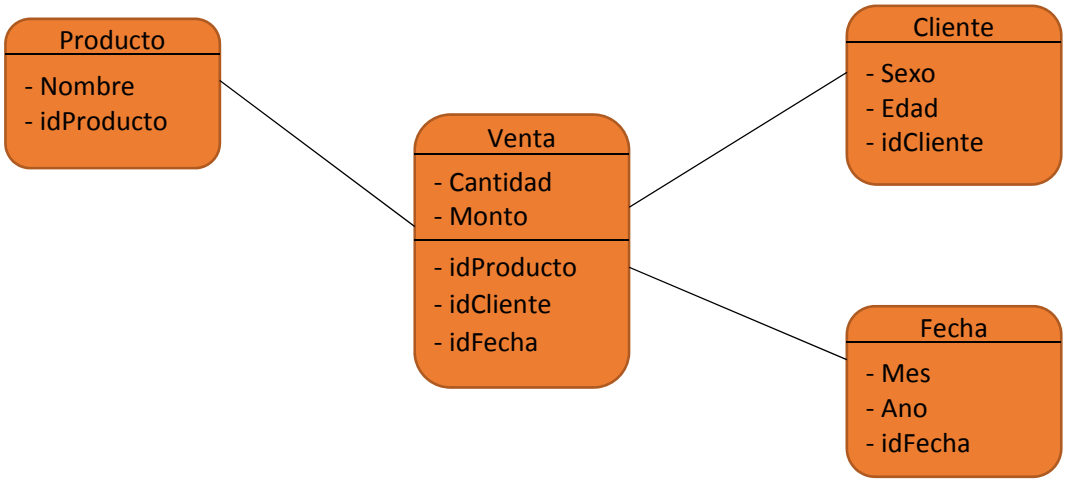

Figura 2.4: Modelo multidimensional para el almacén de datos de las ventas en una tienda

Después de definir todos los elementos que componen la ontología se presentan cada una de las entidades y relaciones con su significado y su transformación al lenguaje de ontologías OWL2. Se considera agregar un atributo a cada declaración

de entidades llamado “tag” para simplificar su nombre y facilitar el procesamiento de estas relaciones en los sistemas de representación.

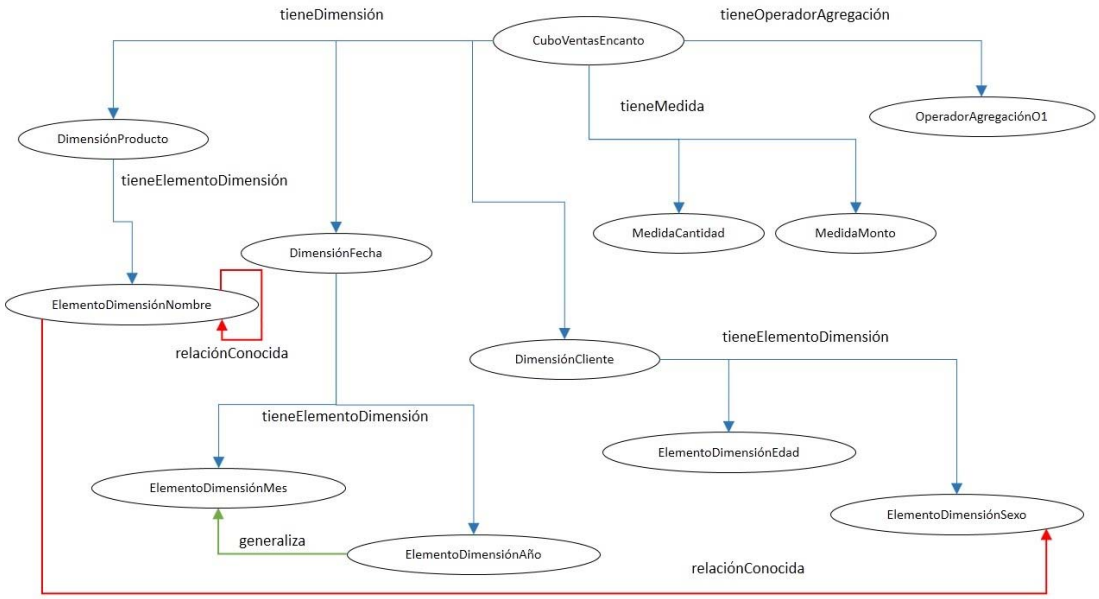

Figura 2.5: Relación entre los elementos de la ontología del dominio VentasEncanto

Declaración de entidades en OWL2

Cubo de datos:

La primera de las entidades que se declara en la ontología “core” es Cubo que representa el concepto cubo de datos que contiene todas los hechos, medidas, dimensiones, elementos de dimensión y operadores de agregación. En su extensión a la ontología de dominio se instancia este concepto con el DataCubo VentasEncanto cuya transformación al lenguaje OWL2 consta de la declaración de una clase cuyo identificador comienza por el símbolo de numeral (#), seguido de la palabra Cube y del nombre del cubo de datos como se muestra a continuación:

```
<Declaration>
 <Class IRI='#Cube_VentasEncanto' tag='VentasEncanto' />
</Declaration>
```

Dimensión:

La entidad Dimensión representa cada elemento del conjunto de dimensiones que contiene el cubo de datos. Primero se declara una clase de propósito general llamada Dimension y para cada dimensión del dominio se declara una especificación de ella. En su declaración consta una clase llamada Dimensión, para la entidad genérica, cuyo identificador comienza por el símbolo de numeral (#), seguido de la palabra Dimension como se muestra a continuación:

```
<Declaration>
 <Class IRI='#Dimension'/>
</Declaration>
```

En el caso de las entidades específicas del conjunto de dimensiones del dominio se extiende esta, declarando una clase cuyo identificador comienza por el símbolo de numeral (#), seguido de la palabra Dimension y del nombre de la dimensión como se muestra a continuación:

```
<Declaration>
 <Class IRI='#Dimension_Producto' tag='Producto'/>
</Declaration>
```

En este punto es válido recordar que con la ontología se define un lenguaje para que sea entendido por una computadora, por esta razón es necesario hacer explícito algo que para el hombre es intuitivo, es el caso en el que la dimensión específica representa una subclase de la clase dimensión. Igualmente este es el punto de extensión de la ontología “core” que tiene el concepto Dimensión con la ontología de dominio que tiene la dimensión específica Producto. Por lo tanto esta especialización de la entidad Dimensión es de vital importancia declararla y se hace de la manera siguiente:

```
<SubClassOf>
 <Class IRI='#Dimension_Producto' tag='Producto'/>
 <Class IRI='#Dimension'/>
</SubClassOf>
```

En esta transformación se establece la generalización de la entidad específica con la entidad dimensión y nótese que es importante el orden debido a que si se intercambian las clases, la relación entre ellas no es de subclase sino de superclase. Un caso similar sucede con las propiedades que se describen más adelante que no son reflexivas y por lo tanto no se puede invertir la relación.

ElementoDimensión:

La entidad ElementoDimensión representa cada elemento del conjunto de atributos que contiene la dimensión. Primero se declara una clase de propósito general llamada DimensionElement y para cada atributo del dominio se declara una especificación de ella. En su declaración consta una clase llamada DimensionElement, para la entidad genérica, cuyo identificador comienza por el símbolo de numeral (#), seguido de la palabra DimensionElement como se muestra a continuación:

```
<Declaration>
 <Class IRI='#DimensionElement'/>
</Declaration>
```

Para el caso de cada atributo específico dentro de cada dimensión se declara una clase cuyo identificador comienza por el símbolo de numeral (#), seguido de la palabra DimensionElement, el nombre del atributo o elemento de dimensión y se incluye el nombre de la dimensión a que pertenece este elemento. La razón de incluir el último es brindar al identificador de la clase toda la información de la ubicación del elemento, así también se resuelve el problema de la duplicidad de nombres en caso de que existan atributos con el mismo nombre en diferentes dimensiones. La transformación se muestra a continuación:

```
<Declaration>
 <Class IRI='#DimensionElement_Sexo_Cliente' tag='Sexo'/>
</Declaration>
```

Igualmente en este caso es necesario establecer la extensión al dominio del elemento de dimensión. Esta especialización se hace de la manera siguiente:

```
<SubClassOf>
 <Class IRI='#DimensionElement_Sexo_Cliente' tag='Sexo'/>
 <Class IRI='#DimensionElement'/>
</SubClassOf>
```

Análogamente al caso de la generalización entre la dimensión específica dentro del conjunto de dimensiones y la clase Dimensión sucede con las clases ElementoDimensión general y específica.

Individuos:

Los individuos en el lenguaje de ontologías representan los valores que puede tomar una entidad, en esta investigación se usan para representar los valores que tiene cada elemento de dimensión. Aunque los individuos no son clases, sí pertenecen a estas, se declaran, y se establece su pertenencia a cada clase. Su declaración es similar a las presentadas anteriormente, se nombra cada individuo usando un identificador con una estructura que comienza por el símbolo de numeral (#), seguido del valor del atributo, después se escribe el nombre del elemento de dimensión al que pertenece y finalmente el nombre de la dimensión a la que pertenece este atributo. Es importante aclarar que esta estructura que siguen los identificadores no establece relación ninguna entre los elementos de la ontología, para establecer estas relaciones se define una aserción entre el individuo y la clase a que pertenece. La declaración es la que se muestra a continuación.

```
<Declaration>
 <NamedIndividual IRI='#Female_Sexo_Cliente' tag='Female'/>
</Declaration>
```

Después de declarado este individuo se pasa escribir la aserción que lo vincula con la clase a que pertenece. Esta se escribe en OWL2 de la forma siguiente:

```
<ClassAssertion>
 <Class IRI='#DimensionElement_Sexo_Persona' tag='Sexo'/>
 <NamedIndividual IRI='#Female_Sexo_Persona' tag='Female'/>
</ClassAssertion>
```

Declaración de relaciones entre entidades en OWL2

Relación “tieneDimensión”:

La primera de las relaciones o propiedades sobre objetos de la ontología que se declara es la pertenencia de una dimensión al cubo de datos y de manera similar a la generalización y especialización de las clases, las relaciones también siguen este formato. La declaración de la propiedad sobre objetos general está conformada por un identificador que comienza por el símbolo de numeral (#), seguido de la frase “hasDimension” como se presenta a continuación:

```
<Declaration>
 <ObjectProperty IRI='#hasDimension'/>
</Declaration>
```

La especialización de esta relación para lograr su extensión a las relaciones del dominio se declara usando un identificador que comienza por el símbolo de numeral (#), seguido del nombre del cubo de datos, la frase “hasDimension” y finalmente el nombre de la dimensión como se hace a continuación:

```
<Declaration>
 <ObjectProperty IRI='#VentasEncanto_hasDimension_Cliente'/>
</Declaration>
```

Se vuelve a enfatizar que esta relación no es reflexiva, por tanto el dominio y el rango de esta relación, definidos más adelante debe hacer referencia, el primero a un cubo de datos y el segundo a una dimensión. Además es semánticamente incorrecto plantear en el ejemplo anterior que Cliente (un elemento del conjunto de

dimensiones) tiene una dimensión que se llama VentasEncanto (siendo este el cubo de datos).

Después de enunciar estas dos relaciones anteriores se pasa a describir su jerarquía. En este caso se declara que la relación particular del dominio es una especialización de la relación “tieneDimensión” de la ontología “core” y se escribe en OWL2 de la forma siguiente:

```
<SubObjectPropertyOf>
 <ObjectProperty IRI='#VentasEncanto_hasDimension_Cliente’/>
 <ObjectProperty IRI='#hasDimension'/>
</SubObjectPropertyOf>
```

A continuación se define el Dominio de la propiedad “tieneDimensión”. En esta definición intervienen la relación mencionada y la clase Cubo para definir que el conjunto de partida o dominio de esta propiedad sobre objetos es el cubo de datos. Se escribe de la manera siguiente:

```
<ObjectPropertyDomain>
 <ObjectProperty IRI='#VentasEcanto_hasDimension_Cliente’/>
 <Class IRI='#Cube_VentasEncanto’/>
</ObjectPropertyDomain>
```

Ahora se define el Rango de la propiedad “tieneDimensión”. En esta definición intervienen la relación mencionada y la clase Dimensión para definir que el conjunto de llegada o imagen de esta propiedad sobre objetos es una dimensión. Se escribe del modo siguiente:

```
<ObjectPropertyRange>
 <ObjectProperty IRI='#VentasEncanto_hasDimension_Cliente'/>
 <Class IRI='#Dimension_Cliente' tag='Cliente'/>
</ObjectPropertyRange>
```

Relación “tieneElementoDimensión”:

La siguiente propiedad cumple igualmente con todas las definiciones presentadas anteriormente con la única diferencia que se establece entre la dimensión y el elemento de dimensión.

La declaración de la propiedad “tieneElementoDimensión” de propósito general está conformada por un identificador que comienza por el símbolo de numeral (#) seguido de la frase “hasDimensionElement” como se muestra a continuación:

```
<Declaration>
 <ObjectProperty IRI='#Cliente_hasDimensionElement_Sexo'/>
</Declaration>
```

La especificación de esta relación es muy similar a la de la propiedad “tieneDimensión” y la estructura del identificador comienza por el símbolo de numeral (#), a continuación el nombre de la dimensión, la frase “hasDimensionElement” y finalmente el nombre del elemento de dimensión del modo siguiente:

```
<Declaration>
 <ObjectProperty IRI='#Cliente_hasDimensionElement_Sexo'/>
</Declaration>
```

También se mantiene la extensión entre esta propiedad del dominio y su generalización descrita como una subpropiedad. Este ejemplo ilustra su estructura:

```
<SubObjectPropertyOf>
 <ObjectProperty IRI='#Cliente_hasDimensionElement_Edad'/>
 <ObjectProperty IRI='#hasDimensionElement'/>
</SubObjectPropertyOf>
```

Para la definición del Dominio de la propiedad “tieneElementoDimensión” se incluye la propia relación y la clase Dimensión específica para precisar que el conjunto de partida o dominio de esta propiedad sobre objetos es la dimensión. Se escribe del modo siguiente:

```
<ObjectPropertyDomain>
 <ObjectProperty IRI='#Cliente_hasDimensionElement_Sexo'/>
 <Class IRI='#Dimension_Persona'/>
</ObjectPropertyDomain>
```

En la definición del Rango de la propiedad “tieneElementoDimensión” interviene la relación mencionada y la clase ElementoDimensión para definir que el conjunto de llegada o imagen de esta propiedad sobre objetos es un elemento de dimensión. Se escribe de la manera siguiente:

```
<ObjectPropertyRange>
 <ObjectProperty IRI='#Cliente_hasDimensionElement_Sexo'/>
 <Class IRI='#DimensionElement_Sexo_Cliente' tag='Sexo'/>
</ObjectPropertyRange>
```

Relación “tieneRelaciónConocida”:

Por último se presenta la relación “tieneRelaciónConocida”, esta es la más importante de las asociaciones entre elementos pues es la que permite a los expertos del negocio introducir el conocimiento previo que poseen para enriquecer el dominio.

Para declarar la relación anterior se realiza de la misma manera que las precedentes. Su identificador comienza por el símbolo de numeral (#) seguido de la frase “hasKnownRelation” como se presenta a continuación:

```
<Declaration>
 <ObjectProperty IRI='#hasKnownRelation'/>
</Declaration>
```

La especialización de esta relación es similar a las anteriores, pero se debe tener en cuenta que este tipo de relación puede involucrar tanto a elementos de dimensión como individuos, por lo que se presta especial interés al dominio y al rango para obtener la semántica correcta y deseada. Se declara usando un identificador que comienza por el símbolo de numeral (#), seguido del nombre del primer elemento de la relación, a continuación la frase “hasKnownRelation” y finalmente el nombre del segundo elemento que tenga relación como se presenta a continuación:

```
<Declaration>
 <ObjectProperty
IRI='#Pan_Nombre_Producto_hasKnownRelation_Leche_Nombre_Product
o'/>
</Declaration>
```

En este caso, el experto establece que de la dimensión Producto, el elemento Nombre con el individuo Pan tiene una relación que se conoce con la dimensión Producto en su elemento de dimensión Nombre y su individuo Leche. Para describir con más claridad esta relación, el experto quiere expresar que generalmente al hacer una venta en la que esté involucrado el producto de nombre Pan también lo está el producto de nombre Leche.

Esta relación tampoco es reflexiva, por tanto el dominio y el rango de esta, definidos más adelante son los que otorgan a la relación su significado semántico.

Después de declarar estas dos relaciones anteriores se pasa a describir su jerarquía. En este caso se describe que la relación particular es una especialización de la relación “tieneRelaciónConocida” y se escribe en OWL2 de la manera siguiente:

```
<SubObjectPropertyOf>
 <ObjectProperty
IRI='#Pan_Nombre_Producto_hasKnownRelation_Leche_Nombre_Product
o'/>
 <ObjectProperty IRI='#hasKnownRelation'/>
</SubObjectPropertyOf>
```

A continuación se define el Dominio de la propiedad “tieneRelaciónConocida”. En esta definición intervienen dos elementos, para los ejemplos anteriores se han usado dos individuos, por lo tanto en este caso también se usan y se escribe de la forma siguiente:

```
<ObjectPropertyDomain>
 <ObjectProperty
IRI='#Pan_Nombre_Producto_hasKnownRelation_Leche_Nombre_Product
o'/>
 <NamedIndividual IRI='#Pan_Nombre_Producto' tag='Pan'/>
```

```
</ObjectPropertyDomain>
```

A continuación se define el Rango. Esta definición es similar a la del dominio y se escribe del modo siguiente:

```
<ObjectPropertyRange>
 <ObjectProperty
IRI='#Pan_Nombre_Producto_hasKnownRelation_Leche_Nombre_Product
o'/>
 <NamedIndividual IRI='#Leche_Nombre_Producto' tag='Leche'/>
</ObjectPropertyRange>
```

2.4 Conclusiones parciales

En este capítulo se estableció un modelo teórico para la representación de asociaciones en el modelo multidimensional, seleccionándose los elementos esenciales que lo componen para asociarle una representación ontológica, igualmente se determinó un mecanismo para la representación ontológica del conocimiento previo sobre asociaciones de elementos en cubos de datos. Al reunir todos estos elementos se definió una ontología “core” para la representación del modelo multidimensional y un mecanismo para obtener la representación ontológica de un cubo de datos.

Capítulo 3
Validación de la solución

Introducción

En este capítulo se analizan las tendencias para la evaluación de ontologías y se hace un análisis para aplicarlas a la solución propuesta en el capítulo anterior. Finalmente se selecciona un método que establece cuatro etapas para evaluar la propuesta y se realizan modificaciones para adaptarlo a este fin. Al final de capítulo se muestran dos ejemplos para mostrar el proceso de transformación.

3.1 Propuestas para la evaluación de ontologías

En (Ramos et al. 2009) se explica que aunque se ha profundizado en las metodologías para la creación de ontologías, todavía no se ha considerado con suficiente interés el tema de su evaluación. También se acota que este proceso debe realizarse de manera similar a la evaluación de un componente de software y se debe enfocar en que las definiciones implementen los requerimientos y den respuesta a las preguntas de competencia preestablecidas y también deben modelar lo más exactamente posible el dominio para el cual fueron creadas.

Los autores que tratan este tema Brewster, Obrst, Prozel, Burton-Jones y Brank, citados por (Ramos, Núñez and Casañas 2009) indican que para evaluar la calidad de la ontología es necesario incluir en este proceso cuatro criterios importantes: taxonomía, lenguaje, vocabulario y rendimiento aunque también es un requisito propuesto en estos trabajos que se compare dicha ontología con otra existente.

Para esta propuesta, no existe otra ontología similar para comparar. En este caso (Ramos, Núñez and Casañas 2009) propone cuatro fases para este fin, sin necesidad de establecer analogías entre ontologías afines. Estas fases están en concordancia con los criterios antes mencionados y se relacionan a continuación:

- Fase 1: Uso correcto del lenguaje
    - Validar que el lenguaje cumpla con los estándares de desarrollo ontológicos.
    - Evaluar sintácticamente la ontología en cada fase de su desarrollo.
- Fase 2: Exactitud de la estructura taxonómica
    - Chequeo de inconsistencias, completitud y redundancia de los términos de la taxonomía.
- Fase 3: Validez del vocabulario
    - Identificar, extraer y organizar los términos significativos del dominio a partir de los documentos.
    - Evaluar el vocabulario considerando las medidas de calidad de resultados usando medidas tales como la precisión y el *recall (exhaustividad)*.
- Fase 4: Adecuación a los requerimientos
    - Verificar que las especificaciones de requerimientos se cumplan.
    - Verificar que las respuestas proporcionadas por la ontología a las preguntas de competencia sean correctas y pertinentes.

3.2 Aplicación del método de evaluación a la propuesta

En la propuesta de ontología del modelo multidimensional que se utiliza para representar dominios de conocimiento a través de puntos de extensión, la propia ontología o cualquier dominio representado como extensión de ella, son únicos en un dominio. Por estas razones se decide aplicar las cuatro fases descritas en el epígrafe anterior para evaluar el modelo propuesto.

Antes de describir la aplicación de cada una de las fases, es necesario aclarar que el resultado de esta investigación no es solo una ontología, sino representar dominios de datos como extensiones de los elementos del modelo multidimensional. Siguiendo esta idea, se distingue que la ontología de dominio resultante de aplicar el proceso de transformación a un almacén de datos será la representación ontológica de este. Lo planteado hace extremadamente difícil la aplicación de cualquier método de evaluación a la propuesta porque se necesitaría evaluar cada dominio representado y es imposible contar con expertos en cada uno de ellos y realizar el proceso a cada almacén que se transforme. Aun así, se propone por parte del autor, realizar la evaluación de la ontología “core” en la que se utilizan los elementos y relaciones del modelo multidimensional y propagar esta validez a cualquier dominio representado usando extensiones de estas relaciones. En este caso si existiese alguna inconsistencia en la información contenida en la ontología de dominio será el resultado de un mal diseño del almacén de datos y no un error de diseño de la ontología resultante.

Después de presentar estas particularidades se describe la forma en que cada una de las fases se aplica a la ontología “core”:

Fase 1: Uso correcto del lenguaje

Uno de los elementos que aporta calidad a la ontología “core” en esta fase es que para su codificación se usa el lenguaje OWL (Ontology Web Languaje), estándar reconocido por la W3C. El otro elemento que también sigue este propósito es el uso en cada paso del marco de chequeo que provee Protégé-OWL en su editor para corregir inconsistencias sintácticas y obtener un código libre de errores. La muestra de este paso se corrobora en la Figura 3.1:

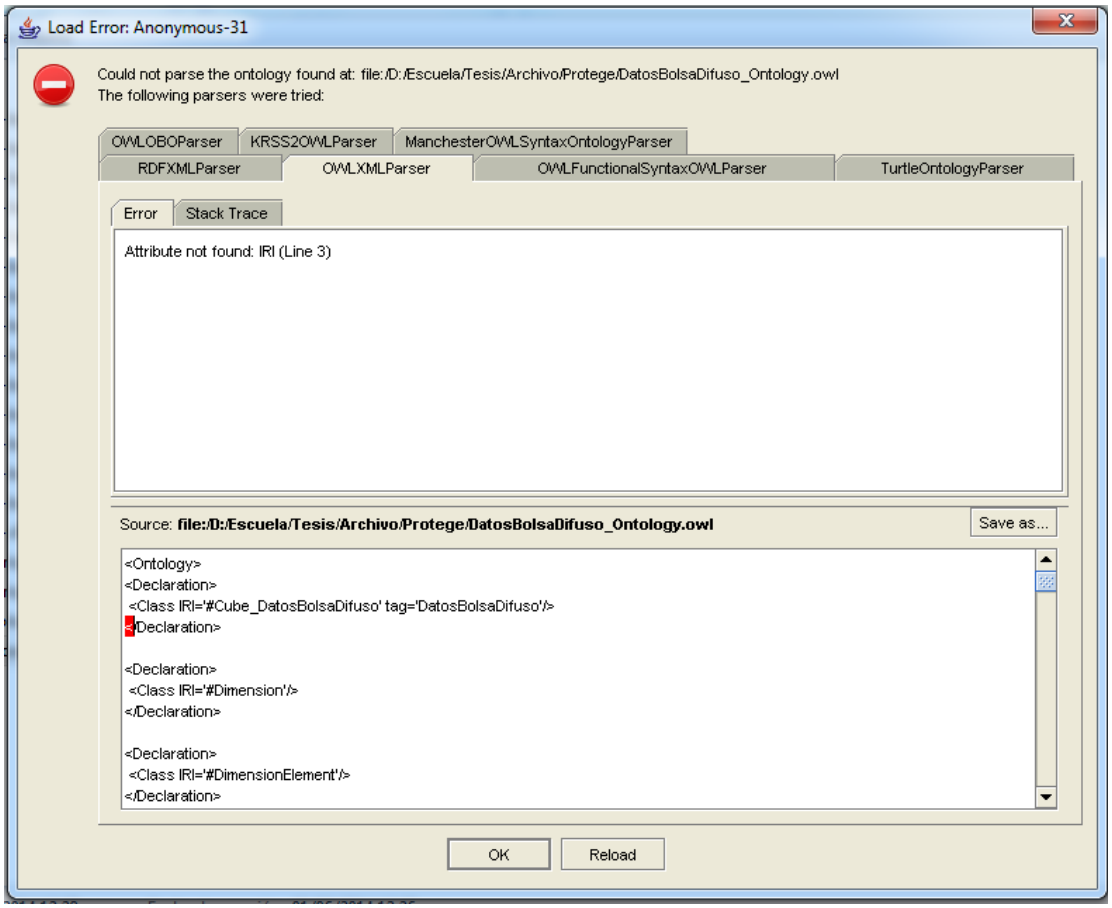

Figura 3.6: Editor de Protégé-OWL para realizar análisis de sintaxis

Fase 2: Exactitud de la estructura taxonómica

En esta fase las actividades que se llevan a cabo para incrementar la calidad de la ontología del modelo multidimensional son: identificación de inconsistencias, completitud de los conceptos y existencia de redundancia en las clases y relaciones. Estas tres actividades se llevaron a cabo haciendo un chequeo manual de cada uno de conceptos del modelo multidimensional y de las relaciones entre estos. El resultado de aplicar esta fase corroboró que todos los conceptos y relaciones se encuentran debidamente ubicados y que no existen inconsistencias.

Fase 3: Validez del vocabulario

Esta es la fase que mayor información aporta en el proceso de evaluación pues se utilizan dos medidas numéricas de cubrimiento para determinar la completitud y validez del vocabulario. Para ambas medidas es necesario que se cuenten los términos significativos, ya sea de manera manual o automatizada, del *corpus* (conjunto más extenso de datos) y de la ontología ($CCorp$ y $COonto$ respectivamente), en este caso se hizo el conteo de forma manual.

La primera de las medidas es la precisión y se calcula usando la fórmula 3:

$$Precisión = CO - C/COnto \qquad [3]$$

Donde:

$CO - C$: Cantidad de términos que se solapan entre la ontología y el *corpus*

Se toman los elementos presentes en la Figura 2.1 como el *corpus* con una cantidad de términos igual a 5 ($CCorp = 5$). Para este paso no se incluyen las relaciones pues no representan conceptos del modelo multidimensional sino asociaciones entre ellos. En la explicación de esta figura se indica que hay una representación ontológica para cada uno de estos elementos por lo tanto la cantidad de términos de la ontología es también 5 ($COnto = 5$) y como en ambos lugares los conceptos se solapan $COnto = CCorp = CO - C = 5$.

Sustituyendo en la fórmula 3 se tiene lo siguiente:

$$Precisión = \frac{5}{5} = 1 \qquad [3.1]$$

Para una precisión del 100%

La segunda de las medidas es el *recall* y se calcula usando la fórmula siguiente:

$$Recall = CO - C/CCorp \qquad [4]$$

Donde:

$CO - C$: Cantidad de términos que se solapan entre la ontología y el *corpus*

En este caso se usan los mismos valores y explicaciones que se usan en la Precisión.

Sustituyendo en la fórmula 4 se tiene lo siguiente:

$$Recall = \frac{5}{5} = 1 \qquad [4.1]$$

Para un *recall* o exhaustividad del 100%

Fase 4: Adecuación a los requerimientos

Para esta fase se proponen dos actividades, la primera de ellas es la verificación de las especificaciones presentadas para la ontología, ya sea en la representación de cada uno de los elementos del modelo multidimensional, las relaciones entre ellos y cada uno de los formalismos para codificar los identificadores y etiquetas, como por ejemplo:

1. Los identificadores deben empezar por el símbolo de numeral (#).
2. Cada identificador debe mostrar la localización dentro de cada elemento del modelo multidimensional, por ejemplo para un ElementoDimensión (ED) se usará un identificador con la estructura #ElementoDimensión_NombreED_NombreDimensión.
3. Incluir un atributo “tag” a las etiquetas que denoten conceptos del modelo multidimensional para incluir su nombre y facilitar el manejo a los sistemas de representación visual de este tipo de ontologías.

La otra actividad es el chequeo conjuntamente con expertos de recorridos sobre la ontología para verificar que el conocimiento representado permita responder

preguntas de competencia[4]. En este caso se restringe el uso de esta actividad pues las únicas preguntas que pueden hacerse son las relacionadas con los elementos genéricos pues se evalúa la ontología “core” o sea si el elemento genérico Dimensión pertenece al elemento genérico Cubo y así sucesivamente. Como se plantea desde el inicio del capítulo cada dominio que se represente a partir de un almacén de datos que se transforme, necesita un experto para formular sus preguntas de competencia y esto, como se ha señalado, no es posible.

Después de ejecutar cada una de estas fases se puede asegurar que la ontología del modelo multidimensional es válida y que cualquier dominio que tenga su representación ontológica a partir de puntos de extensión con la primera también lo es, siempre que la estructura de modelado del almacén de datos sea semánticamente correcta.

3.3 Aplicación de la solución a los datacubos del Censo y la Bolsa

Con el fin de ejemplificar el proceso de transformación propuesto se usa un sistema para la implementación de cubos de datos y la ejecución de algoritmos de extracción de reglas de asociación llamado FCubeFactory. Este sistema sigue una arquitectura cliente-servidor y está montado sobre una plataforma web constituida por una aplicación servidora que almacena los cubos de datos implementados y los métodos sobre estos. La parte cliente es donde el usuario puede observar el resultado de sus solicitudes a través de un navegador web. Antes de ejecutar el servidor se debe tomar el código en el lenguaje de programación que se haya implementado y construir su ejecutable. En este caso se usó el lenguaje de programación Java en su versión 1.6.0_20 y el entorno de desarrollo integrado NetBeans en su versión 7.2.1. El archivo ejecutable se construye con la extensión .jar para incluirlo en las bibliotecas del sistema FCubeFactory. Cuando el sistema se encuentra corriendo, en el navegador se visualiza una interfaz como la que se muestra en la Figura 3.1.

[4] Preguntas que se hacen para seguir caminos entre los conceptos de la ontología

Figura 3.7: Interfaz inicial del sistema FCubeFactory

Como dominio de aplicación se han elegido dos modelos multidimensionales que se describen a continuación:

Modelo multimensional Censo

Este cubo de datos contiene los datos referentes a un censo realizado en un determinado país. El modelo cuenta con una tabla de hechos y nueve dimensiones (Estado Civil, Estudios, Persona, Horas de trabajo, Capital Perdido, Capital Ganado, Relación, País y Tipo de trabajo) modeladas en la Figura 3.2 que se muestra a continuación:

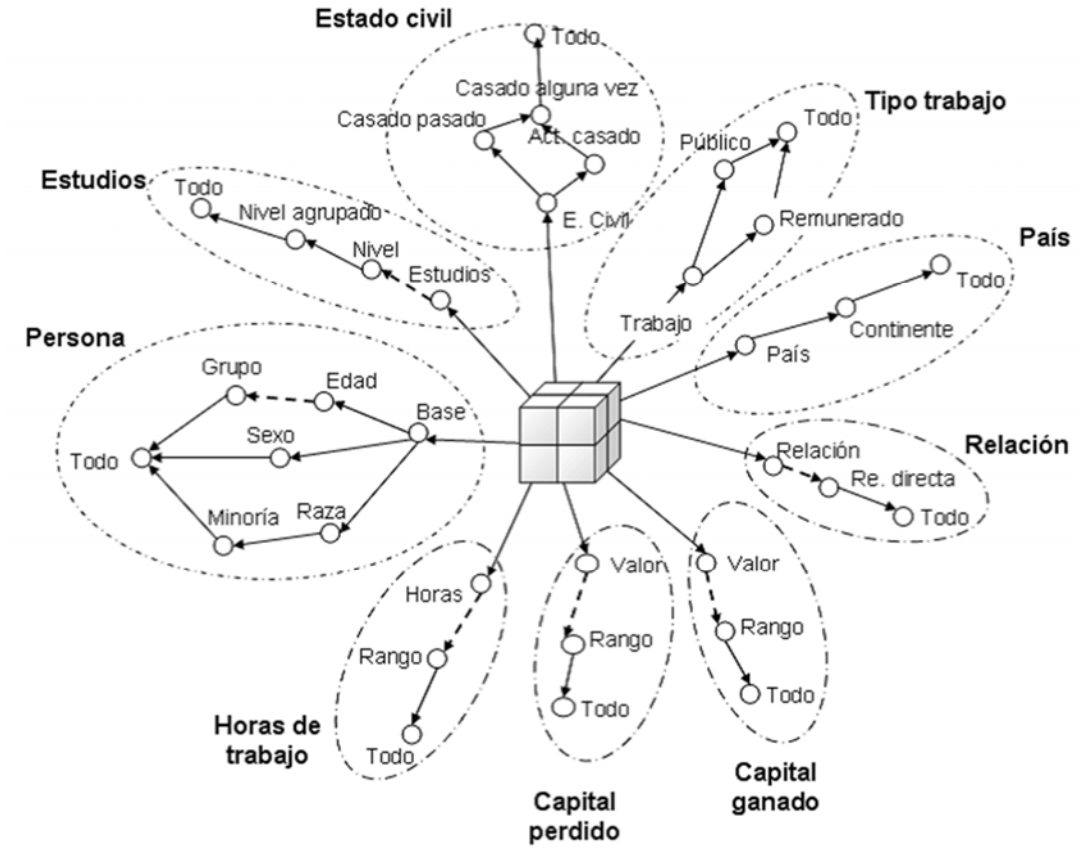

Figura 3.8: Modelo multidimensional del cubo de datos Censo

Para crear la ontología del modelo multidimensional a partir del cubo de datos Censo se hace clic en el botón crear resumen, después se pasa a otra vista en la que se escribe el nombre de esta operación, se elige el cubo de datos CensoCrisp y finalmente se selecciona el método que en este caso es Ontología. En la vista siguiente se entran los parámetros relacionados con el soporte, la complejidad, la función de complejidad y otros. Finalmente se hace click en continuar y se genera el archivo owl con el código de la ontología correspondiente a este almacén.

Figura 3.9: Selección del método Ontología sobre el cubo de datos Censo en la interfaz del FCubeFactory

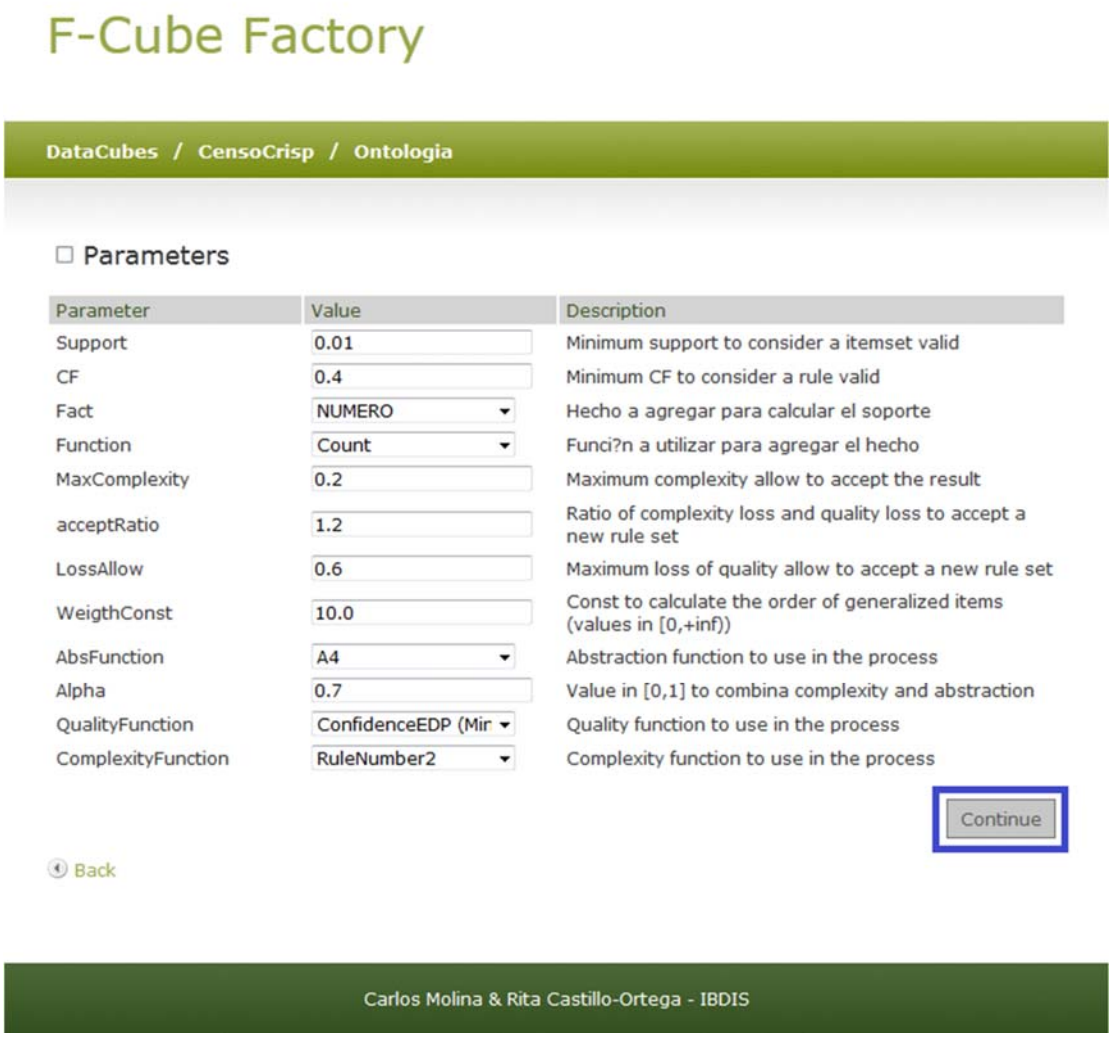

Figura 3.10: Entrada de parámetros para el método Ontología

Al ejecutarse se crea en la dirección especificada un archivo con el nombre CensoCrisp_Ontology en el que se encuentran cada una de las transformaciones descritas en los epígrafes anteriores para este modelo multidimensional.

Se muestran fragmentos de la ontología para corroborar lo anteriormente planteado. Primeramente se presenta la declaración del cubo de datos:

```
<Ontology>
<Declaration>
 <Class IRI='#Cube_CensoCrisp' tag='CensoCrisp'/>
</Declaration>
```

Más adelante la declaración de las dimensiones, sus elementos, individuos y las asociaciones entre ellos:

```
<Declaration>
 <Class IRI='#Dimension_Persona' tag='Persona'/>
</Declaration>

<SubClassOf>
 <Class IRI='#Dimension_Persona' tag='Persona'/>
 <Class IRI='#Dimension'/>
</SubClassOf>

<Declaration>
 <ObjectProperty IRI='#CensoCrisp_hasDimension_Persona'/>
</Declaration>

<Declaration>
 <Class IRI='#DimensionElement_Sexo_Persona' tag='Sexo'/>
</Declaration>

<SubClassOf>
 <Class IRI='#DimensionElement_Sexo_Persona' tag='Sexo'/>
 <Class IRI='#DimensionElement'/>
</SubClassOf>


<Declaration>
 <NamedIndividual IRI='#Female_Sexo_Persona' tag='Female'/>
</Declaration>

<ClassAssertion>
 <Class IRI='#DimensionElement_Sexo_Persona' tag='Sexo'/>
 <NamedIndividual IRI='#Female_Sexo_Persona' tag='Female'/>
</ClassAssertion>

<Declaration>
 <NamedIndividual IRI='#Male_Sexo_Persona' tag='Male'/>
</Declaration>

<ClassAssertion>
 <Class IRI='#DimensionElement_Sexo_Persona' tag='Sexo'/>
 <NamedIndividual IRI='#Male_Sexo_Persona' tag='Male'/>
</ClassAssertion>
```

```
<Declaration>
 <ObjectProperty IRI='#Persona_hasDimensionElement_Sexo'/>
</Declaration>


<SubObjectPropertyOf>
 <ObjectProperty IRI='#Persona_hasDimensionElement_Sexo'/>
 <ObjectProperty IRI='#hasDimensionElement'/>
</SubObjectPropertyOf>

<ObjectPropertyDomain>
 <ObjectProperty IRI='#Persona_hasDimensionElement_Sexo'/>
 <Class IRI='#Dimension_Persona'/>
</ObjectPropertyDomain>

<ObjectPropertyRange>
 <ObjectProperty IRI='#Persona_hasDimensionElement_Sexo'/>
 <Class IRI='#DimensionElement_Sexo_Persona' tag='Sexo'/>
</ObjectPropertyRange>
```

Al final de este archivo el experto del conocimiento del dominio introduce las reglas del conocimiento previo para enriquecer la ontología del almacén.

Tabla 3.1 Resumen de aplicar las transformaciones al DataCubo Censo

Cantidad de clases	25
Cantidad de instancias	125
Cantidad de relaciones	24

Modelo multimensional Bolsa

Este cubo de datos contiene los datos referentes a un análisis del comportamiento de la bolsa. Cuenta con una tabla de hechos y catorce dimensiones (Puntos, Mercado, Riesgo, Número de transacciones, Volatilidad, Razón de Sharpe, Potencial, Disposición, Ganancia, Consistencia, E01, E04, Confiablidad, Estrategia) modeladas en la Figura 3.5.

Figura 3.11: Modelo multidimensional del cubo de datos Bolsa

Para crear la ontología de este modelo multidimensional a partir del cubo de datos Bolsa se hace clic en el botón crear resumen, después se pasa a otra vista en la que se escribe el nombre de esta operación, se elige el cubo de datos BolsaDifuso y finalmente se selecciona el método que en este caso es Ontología. En la vista siguiente se entran los parámetros relacionados con el soporte, la complejidad, la función de complejidad y otros. Finalmente se hace click en continuar y se genera el archivo owl con el código de la ontología correspondiente a este almacén.

Figura 3.12: Selección del método Ontología sobre el cubo de datos Bolsa en la interfaz del FCubeFactory

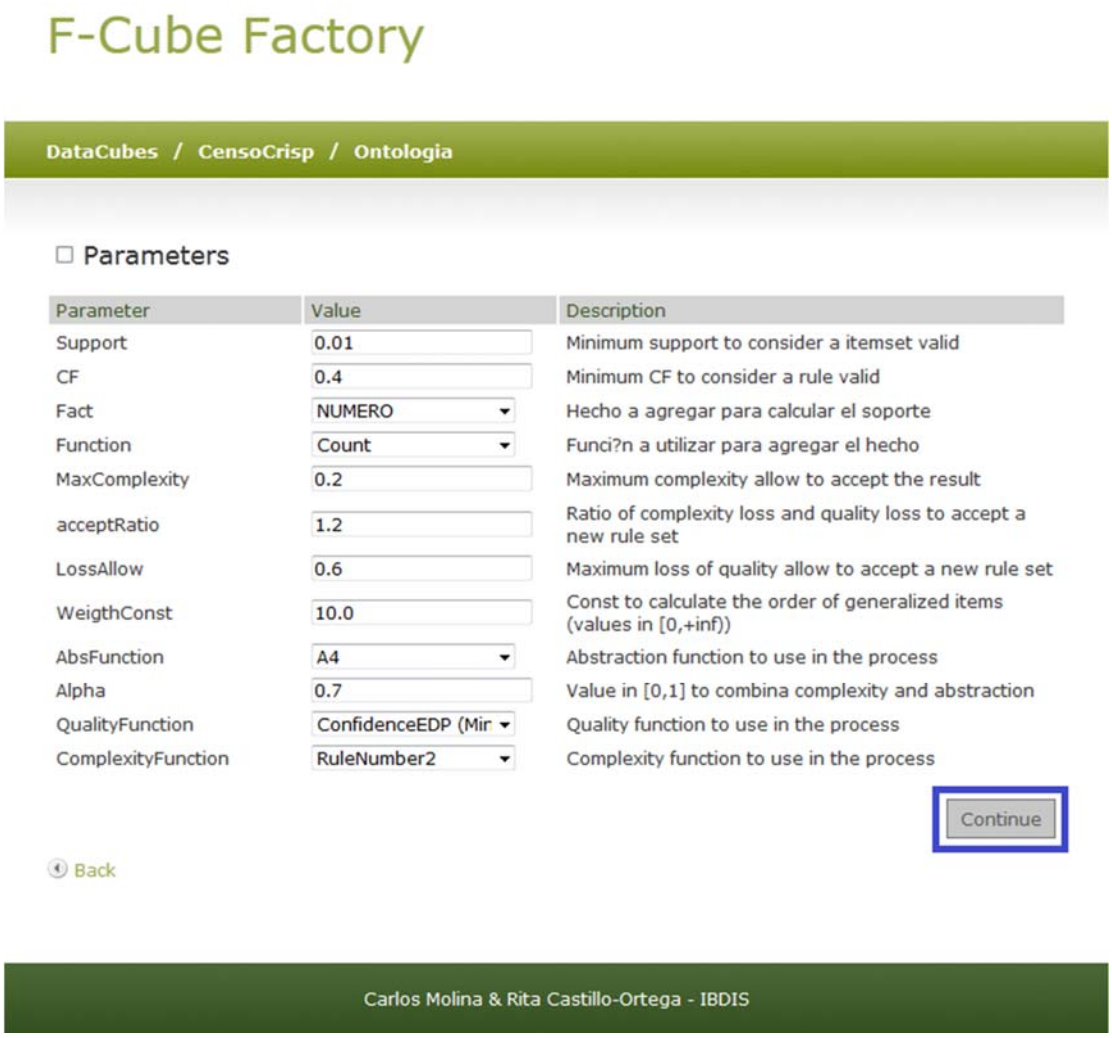

Figura 3.13: Entrada de parámetros para el método Ontología

Al ejecutarse se crea en la dirección especificada un archivo con el nombre BolsaDifuso_Ontology en el que se encuentran cada una de las transformaciones descritas en los epígrafes anteriores para este modelo multidimensional.

Se muestran fragmentos de la ontología para corroborar lo anteriormente planteado. Primeramente se presenta la declaración del cubo de datos:

```
<Ontology>
<Declaration>
 <Class IRI='#Cube_BolsaDifuso' tag='CensoDifuso'/>
</Declaration>
```

Más adelante la declaración de las dimensiones, sus elementos, individuos y las asociaciones entre ellos:

```
<Declaration>
 <Class IRI='#Dimension_Ganancia' tag='Ganancia'/>
</Declaration>

<SubClassOf>
 <Class IRI='#Dimension_Ganancia' tag='Ganancia'/>
 <Class IRI='#Dimension'/>
</SubClassOf>

<Declaration>
 <ObjectProperty IRI='#DatosBolsaDifuso_hasDimension_Ganancia'/>
</Declaration>

<Declaration>
 <Class IRI='#DimensionElement_Calidad_Ganancia' tag='Calidad'/>
</Declaration>

<SubClassOf>
 <Class IRI='#DimensionElement_Calidad_Ganancia' tag='Calidad'/>
 <Class IRI='#DimensionElement'/>
</SubClassOf>

<Declaration>
 <NamedIndividual IRI='#1. Bueno_Calidad_Ganancia' tag='1. Bueno'/>
</Declaration>

<ClassAssertion>
 <Class IRI='#DimensionElement_Calidad_Ganancia' tag='Calidad'/>
 <NamedIndividual IRI='#1. Bueno_Calidad_Ganancia' tag='1. Bueno'/>
</ClassAssertion>

<Declaration>
 <NamedIndividual IRI='#2. Normal_Calidad_Ganancia' tag='2. Normal'/>
</Declaration>

<ClassAssertion>
 <Class IRI='#DimensionElement_Calidad_Ganancia' tag='Calidad'/>
 <NamedIndividual IRI='#2. Normal_Calidad_Ganancia' tag='2. Normal'/>
</ClassAssertion>

<Declaration>
 <NamedIndividual IRI='#3. Malo_Calidad_Ganancia' tag='3. Malo'/>
</Declaration>
```

```
<ClassAssertion>
 <Class IRI='#DimensionElement_Calidad_Ganancia' tag='Calidad'/>
 <NamedIndividual IRI='#3. Malo_Calidad_Ganancia' tag='3. Malo'/>
</ClassAssertion>

<Declaration>
 <ObjectProperty IRI='#Ganancia_hasDimensionElement_Calidad'/>
</Declaration>

<SubObjectPropertyOf>
 <ObjectProperty IRI='#Ganancia_hasDimensionElement_Calidad'/>
 <ObjectProperty IRI='#hasDimensionElement'/>
</SubObjectPropertyOf>

<ObjectPropertyDomain>
 <ObjectProperty IRI='#Ganancia_hasDimensionElement_Calidad'/>
 <Class IRI='#Dimension_Ganancia'/>
</ObjectPropertyDomain>

<ObjectPropertyRange>
 <ObjectProperty IRI='#Ganancia_hasDimensionElement_Calidad'/>
 <Class IRI='#DimensionElement_Calidad_Ganancia' tag='Calidad'/>
</ObjectPropertyRange>

<ObjectPropertyRange>
 <ObjectProperty IRI='#DatosBolsaDifuso_hasDimension_Ganancia'/>
 <Class IRI='#Dimension_Ganancia' tag='Ganancia'/>
</ObjectPropertyRange>
```

Aquí continúa con la declaración de la próxima dimensión después de especificar todas las relaciones, su dominio y rango.

```
<Declaration>
 <Class IRI='#Dimension_Serie Perdida' tag='Serie Perdida'/>
</Declaration>
```

Finalmente antes de cerrar la ontología con la etiqueta </Ontology> se declaran las reglas del conocimiento previo para enriquecer la ontología.

Tabla 3.2 Resumen de aplicar las transformaciones al DataCubo Bolsa

Cantidad de clases	23
Cantidad de instancias	39
Cantidad de relaciones	22

3.3 Resultados y discusión

Con el avanzado desarrollo de las Tecnologías de la Información y las Comunicaciones (TICs) y su significativo aporte a la sociedad del conocimiento se han elevado los estándares para la información que se usa en el proceso de toma de decisiones. Los almacenes de datos contribuyen a este propósito eliminando sesgos de interpretaciones empíricas y brindando todos los elementos posibles a los decisores para elegir la alternativa correcta. Se valora también el conocimiento que puedan proveer los expertos del dominio a tratar en función de elevar la eficiencia de este proceso al lograr obtener conocimiento valioso con la menor cantidad de reglas posibles. De igual forma resulta un procedimiento eficaz al proveer la capacidad para extraer y presentar información precisa a partir de lo contenido en el almacén.

De manera similar al alcance de los sistemas de bases de datos que quedó limitado cuando surgió el término OLAP y fue necesario introducir los almacenes de datos, también es necesario incluir el proceso de minería a los almacenes de datos con el propósito de extraer conocimiento sobre estos sistemas de apoyo a la toma de decisiones. Se plantea por parte del autor que una buena práctica en esta extracción es introducir conocimiento previo que posee un experto del dominio a tratar, en forma de relaciones, para simplificar el resultado final y que sea más entendible para su análisis.

En el epígrafe anterior se utilizan dos cubos de datos para realizar la transformación del modelo multidimensional a su correspondiente representación ontológica. El primero de estos almacenes tiene el nombre CensoCrisp y sus nueve dimensiones

son Estado Civil, Estudios, Persona, Horas de trabajo, Capital Perdido, Capital Ganado, Relación, País y Tipo de trabajo que se recogen en la Figura 3.2. El segundo, de nombre BolsaDifuso con las dimensiones Puntos, Mercado, Riesgo, Número de transacciones, Volatilidad, Razón de Sharpe, Potencial, Disposición, Ganancia, Consistencia, E01, E04, Confiablidad, Estrategia, para un total de catorce, modeladas en la Figura 3.5. El propósito de estos almacenes es guardar información del tema al que están orientados, en el primer caso sobre el censo para determinar los ingresos de las personas y el segundo caso sobre los valores de la bolsa para predecir su comportamiento. De esta forma se brinda soporte a los directivos para anunciar tendencias y tomar decisiones.

Por otro lado las ontologías resultantes de aplicar los procesos de transformación, descritos en el Capítulo 2 de este documento, también almacenan información de dominio relacionada con el tema a que está orientado el almacén, agregando las relaciones conocidas que existan entre sus elementos. En otras palabras se usan para representar los conceptos del dominio como extensiones del modelo multidimensional y la relación que exista entre algunos de ellos para simplificar el proceso de extracción de conocimiento. En este caso se usan etiquetas del lenguaje marcado OWL2 en lugar de tablas, niveles y atributos para representar las entidades y sus relaciones. Esta característica trae a la ontología a un lugar intermedio entre la comprensión de la computadora y la del ser humano pues este último entiende mejor relaciones entre conceptos que pueden ser mostradas usando un sistema de representación que datos en una tabla.

3.4 Conclusiones parciales

En este capítulo se constató la efectividad y validez de la ontología “core” del modelo multidimensional para extenderla a la representación ontológica de un dominio a través de la solución de un caso de estudio. Se demostró el correcto funcionamiento de los mecanismos de transformación con la implementación de una aplicación

orientada a obtener la representación ontológica de un cubo de datos y se llevaron a cabo dos ejemplificaciones que permitieron validar la misma.

Conclusiones

En esta investigación se desarrolló una ontología para representar de forma integrada la estructura de un cubo de datos con sus dimensiones, elementos de dimensión, medidas, hechos y operadores de agregación junto al conocimiento subyacente del dominio que posee un experto del negocio en cuestión, relacionado con las asociaciones entre sus elementos. Para ello se han cumplido los aspectos siguientes:

1. Se estableció un marco conceptual de referencia para soportar los fundamentos teóricos de la investigación y dominar cada aspecto de la propuesta, lo que permitió definir los almacenes de datos y sus elementos como parte del modelo multidimensional. Se introduce la minería de datos como proceso para extraer conocimiento del almacén en forma de reglas de asociación, que también se utilizan para expresar el conocimiento previo. Esto respalda la creación de una ontología que integra la estructura multidimensional del almacén con el conocimiento previo.
2. Se estableció un modelo teórico para la representación de asociaciones entre los elementos del modelo multidimensional, seleccionándose los conceptos esenciales que lo componen para asociarles una representación ontológica.
3. Se definió una ontología de tipo “core” como artefacto para representar los elementos del modelo multidimensional y las asociaciones entre ellos, obteniendo la representación ontológica de un cubo de datos.
4. Se definieron cada uno de los mecanismos para transformar los elementos del modelo multidimensional y el conocimiento previo del dominio para la representación ontológica de un cubo de datos a través de las asociaciones entre sus elementos.

5. Se constató la efectividad y validez de la ontología “core” del modelo multidimensional para extenderla a la representación ontológica de un dominio a través de un caso de estudio. Se demostró el correcto funcionamiento de los mecanismos de transformación con la implementación de una aplicación orientada a obtener la representación ontológica de un cubo de datos y se llevaron a cabo dos ejemplificaciones para validar la misma.

Recomendaciones

- Utilizar esta propuesta, junto a la aplicación de una técnica de minería a los almacenes de datos como complemento a la inteligencia de negocios en función de obtener un elemento integrador que reúna el conocimiento presente en el almacén con el conocimiento subyacente del dominio que posee un experto.
- Utilizar la representación de un dominio a través de la ontología del modelo multidimensional integrado con un sistema de representación donde el experto modele su conocimiento en forma gráfica.
- Utilizar todo el conocimiento contenido en la ontología como entrada para un algoritmo de reducción de reglas de asociación que sea capaz de usar el conocimiento previo para reducir el tamaño y la cantidad de las reglas de asociación obtenidas en el proceso de minería.

Referencias Bibliográficas

- A.MARTIN, M.MANJULA AND P. VENKATESAN A Business Intelligence Model to Predict Bankruptcy using Financial Domain Ontology with Association Rule Mining Algorithm. International Journal of Computer Science, 2011, 8(3).
- AGRAWAL, R., A. GUPTA AND S. SARAWAGI. Modeling multidimensional databases. 1995.
- AGRAWAL, R., T. IMIELINKSI AND A. SWAMI Mining association rule between sets of items in large databases. Proceedings of ACM SIGMOD, 1993, 207–216.
- AGRAWAL, R. AND R. SRITKANT Fast algorithms for mining association rules in large databases. Proceedings of 20th International Conference on Very large data Bases, 1994, 478–499.
- ALHAJJ, R. AND M. KAYA Integrating Fuzziness into OLAP for Multidimensional Fuzzy Association Rules Mining. IEEE Computer Society, Washington, DC, USA, 2003, ICDM '03: Proceedings of the Third IEEE International Conference on Data Mining, 469.
- ANTUNES, C. Pattern Mining over Star Schemas in the Onto4AR Framework. IEEE International Conference on Data Mining Workshops, 2009.
- DAEDALUS Minería de Datos. DAEDALUS – Data, Decisions and Language, 2002.
- DIAZ, J., C. MOLINA AND M. A. VILA Reducción de redundancia en reglas de asociació multinivel 2013.
- DIAZ, J., C. MOLINA AND M. A. VILA A knowledge guided method to reduce redundancy in multilevel association rules 2014.
- DOMÍNGUEZ-DORADO, M. Todo Programación. Editorial Iberprensa, 2005, (13).

- GOSLING, J., B. JOY, G. STEELE AND G. BRACHA The Java language specification. Addison-Wesley, 2005, (Third Edition).
- GUARINO, N. Formal Ontology and Information Systems 6-8 June 1998 1998.
- GUARINO, N., D. OBERLE AND S. STAAB What Is an Ontology? S. Staab and R. Studer, 2009, Chapter 6: Handbook on Ontologies(International Handbooks on Information Systems), 1-17.
- HORROCKS, I., P. F. PATEL-SCHNEIDER AND F. V. HARMELEN From SHIQ and RDF to OWL: The Making of a Web Ontology Language. Journal of Web Semantics, 2003.
- INMON, W. H. AND R. D. HACKATHORN Using the DataWarehouse 1994.
- KIMBALL, R. The Datawarehouse Toolkit. John Wiley & Sons, 1996.
- LAPUENTE, M. J. L. Ontologías. Hipertexto: El nuevo concepto de documento en la cultura de la imagen. Tesis doctoral. Universidad Complutense de Madrid, 2013.
- MARTIN, A., D.MALADHY AND D. V. P. VENKATESAN A FRAMEWORK FOR BUSINESS INTELLIGENCE APPLICATION USING ONTOLOGICAL CLASSIFICATION 2 Feb 2011 2011, 3.
- MARTÍNEZ, A. V. Minería de datos. Una introducción 2003.
- MOLINA, C. Imprecisión e Incertidumbre en el Modelo Multidimensional: Aplicación a la Minería de Datos 2005.
- MOTODA, H. AND K. OHARA Apriori. s.l. Taylor & Francis Group, 2009.
- NEBOT, V., R. BERLANGA, J. M. PÉREZ, M. J. ARAMBURU, et al. MULTIDIMENSIONAL INTEGRATED ONTOLOGIES: A FRAMEWORK FOR DESIGNING SEMANTIC DATA WAREHOUSES 2011.
- OLAPCOUNCIL.
- RAMOS, E., H. NÚÑEZ AND R. CASAÑAS Esquema para evaluar ontologías únicas para un dominio de conocimiento. Enlace, Abril 2009 2009, 6(1).
- RENNOLLS, K. An Intelligent Framework (O-SS-E) for Data Mining, Knowledge Discovery and Business Intelligence. IEEE Computer Society, 2005.

- STUDER, R., R. BENJAMINS AND D. FENSEL Knowledge engineering: Principles and methods. Data & Knowledge Engineering, 1998.
- VALLEJOS, S. J. Minería de Datos. Universidad Nacional del Nordeste. Facultad de Ciencias Exactas, Naturales y Agrimensura, 2006.
- W3C_RECOMMENDATION OWL 2 Web Ontology Language Primer. W3C Recommendation 27 October 2009, 27 October 2009.
- WU, C.-A., W.-Y. LEE, M.-C. TSENG AND C.-C. WU The Research of Ontology-Assisted Data Mining Technology. Journal of Internet Technology, 2007, 8(4).
- XU, J., B. XU AND H. CAI Research and Application of Hybrid-Driven Data Warehouse Modeling Method 2011.
- ZHENG, L., F. LI AND H. GUI The Research of Ontology-Assisted Data Mining Technology 2010.
- ZHU, H. On-Line Analytical Mining of Association Rules 1998.