

Procedimiento para la obtención de un modelo ontológico para representar la información contenida en bases de datos

Alfredo Vega-Ramírez, Irlán Grangel-González, Inty Sáez-Mosquera,
Raúl García-Castro

Ontology Engineering Group, Departamento de Inteligencia Artificial, Escuela
Técnica Superior de Ingenieros Informáticos, Universidad Politécnica de Madrid,
España
alfredo.vega@vertice.cu, irlangrangel2003@gmail.com, intysaez@gmail.com,
rgarcia@fi.upm.es

Resumen El presente trabajo propone un procedimiento a través del cual un razonador evalúa la información de una base de datos y la clasifica de forma automática en conceptos, relaciones, roles y atributos. Esta clasificación se desarrolla mediante un procedimiento dividido en dos métodos: primero, un Algoritmo de migración el cual genera una ontología con los elementos del esquema relacional de la base de datos. El segundo método es la Estrategia de clasificación de la información, esta consiste en una serie de consultas *SPARQL* mediante las que se clasifica la información de la base de datos.

Palabras claves: Esquema Relacional, Algoritmo de Migración, Modelo Ontológico, Estrategia de Razonamiento, Consultas *SPARQL*

Abstract This paper proposes a method by which a reasoner evaluates information from a database and automatically classifies in concepts, relationships, roles and attributes. This classification is developed through a procedure divided into two methods: first, a migration algorithm which generates an ontology with elements of relational schema database. The second method is the strategy classification of information, this is a series of *SPARQL* queries through that classified using the information the database.

Keywords: Relational Schema, Algorithm of Migration, Ontological Model, Strategy of Reasoning, *SPARQL* Queries.

1. Introducción

Actualmente, las tareas de búsqueda, recuperación, reutilización e integración de la información en la *Web* son cada vez más arduas, debido al carácter heterogéneo que presenta el contenido de las fuentes de datos[1]. Esto ha demandado una mayor expresividad y potencia del lenguaje de los métodos de

consultas utilizados tradicionalmente[2]. Estas necesidades tienen su origen en la incapacidad que presentan los modelos de bases de datos para **determinar el significado de la información** contenida en estos modelos.

El presente artículo expone un procedimiento a través del cual se aprovechan los beneficios de las tecnologías semánticas, con el objetivo de obtener un modelo ontológico a través del cual se determinen conceptos, relaciones, roles y atributos de la información almacenada en bases de datos mediante una estrategia de razonamiento. Se utilizará **información** para representar un determinado **conocimiento** de un dominio, en términos de definiciones de conceptos básicos y de las relaciones entre ellos.

El procedimiento emplea un grupo de consultas *SQL* para la obtención de elementos del esquema relacional de la base de datos. Los que son utilizados por un algoritmo de migración para conformar un modelo ontológico estructurado a través de clases, propiedades y sus relaciones, con una capacidad expresiva considerablemente mayor a la del modelo original. Lo anterior permite obtener la información mejor definida y dotada de mayor significado[3].

La clasificación de la información se realiza mediante consultas *SPARQL*, las que modelan las definiciones de conceptos, relaciones, atributos y roles dadas por Nicola Guarino en [4]. Este lenguaje consta de una capacidad semántica, que permite realizar definiciones de conceptos en forma de consultas al mismo nivel del lenguaje, a través del cual se define el modelo ontológico¹.

2. Estado de la cuestión

La *Web* semántica surgió con el propósito de:“exponer en la *Web* la gran cantidad de bases de datos relacionales existentes de modo que puedan ser procesadas automáticamente” [5]. Muchos han sido los sistemas desarrollados para lograr tal propósito, a continuación se realiza un análisis de los diversos sistemas que permiten realizar procesos de migración y/o *upgrade*². De estos se expondrá a *grosso modo* sus características principales.

2.1. *OntoLift*(Universidad de Karlsruhe)

Creado en el contexto del proyecto europeo *WonderWeb*³ y descrito en [6], se trata de un sistema orientado al *upgrade* de la información almacenada en una base de datos relacional(BD) a la *Web* Semántica mediante una ontología a partir del esquema relacional de dicha base.

Los autores definen un proceso de traducción (*lifting process*) que permite convertir las relaciones del esquema relacional en conceptos, atributos y relaciones de la ontología. En esencia se trata de un conjunto de pasos de ingeniería

¹ <http://xsparql.deri.org/>

² Proceso mediante el cual se enriquece semánticamente un modelo existente, obteniendo un nuevo modelo con mayor capacidad expresiva

³ *Ontology Infrastructure for the semantic Web*. IST-2001-33052

inversa para definir una ontología. Una vez definida la ontología, el sistema realiza la extracción de datos del modelo relacional mediante consultas sobre la ontología.

2.2. *Kaon-Reverse*(Universidad de Karlsruhe)

El sistema *Kaon-Reverse*[7][8] permite desarrollar dos tareas fundamentales, en primer lugar la creación de ontologías mediante esquemas de base de datos mediante un proceso de ingeniería inversa y en segundo lugar la definición de correspondencias entre ambos modelos.

Los modelos entre los que se lleva el alineamiento son: una base de datos relacional y una ontología en *RDF* o *DAML + OIL*. En particular de cada modelo se utilizan los siguientes componentes: de la ontología, clases y propiedades y de las bases de datos: tablas, columnas y llaves (primarias y foráneas). En general se trata de un proceso de correspondencias completamente manual, llevado a cabo entre elementos de uno y otro modelo.

2.3. *OntoStudio*(Ontoprise GmbH)

*OntoStudio*⁴, es un *framework* para la gestión de ontologías y aplicaciones basadas en ellas. Este permite llevar a cabo un *upgrade* de bases de datos a ontologías, para facilitar el establecimiento de correspondencias y *mappings* indirectos entre la base de datos y la ontología. Esta indirección es debida a que el modelo relacional es convertido a una ontología mediante el servicio *Database-Schema Import* y es entre esta ontología y cualquier otra, que se establecen las correspondencias.

El proceso de importación del algoritmo es relativamente sencillo, a grandes rasgos consiste, en generar una ontología a partir de modelos relacionales, teniendo en cuenta las equivalencias de los elementos de las bases de datos en las ontologías.

2.4. *D2R-MAP*(Universidad de Berlín)

El sistema *D2R-MAP*[9], permite alinear bases de datos relacionales con ontologías *RDF*. Consta de dos componentes: un lenguaje declarativo de descripción de correspondencias *D2R* y un procesador de dicho lenguaje, capaz de realizar una extracción masiva (proceso *Bath*) del contenido de la base de datos y la generación de instancias de la ontología. Este sistema desarrolla el proceso de descripción de correspondencias de forma manual entre los elementos de ambos modelos.

⁴ <http://www.ontoprise.de/>

2.5. *Ontology Based Query Processing*(Universidad de Humboldt, Berlín)

Descrito en [10], se trata de un enfoque orientado a la recuperación de la información mediante un proceso de mezcla entre la ontología y el modelo relacional. Los autores plantean el uso de ontologías para el procesamiento de consultas sobre un Sistema Gestor de Bases de Datos(SGBD), para lo cual parten de la hipótesis de que si existe una asociación entre la ontología y el modelo relacional, definido en tres tipos de correspondencias o *mappings*:

1. Relations-Concepts: correspondencias entre los conceptos de la ontología y las relaciones del modelo relacional.
2. Atributes-Concepts: correspondencias entre atributos del modelo relacional y los conceptos de la ontología.
3. Values-Concepts: correspondencias entre valores de atributos del modelo relacional y conceptos de la ontología.

De manera general los sistemas antes analizados, no presentan ningún mecanismo para la clasificación de la información almacenada en las bases de datos. Estos están desarrollados principalmente, para conformar una ontología en términos del modelo de base de datos relacional utilizado como origen de datos, realizando un enriquecimiento semántico de la forma en que se representa la información en el modelo original y establecer correspondencias entre ambos modelos. La declaración de estas correspondencias tiene como premisa la presente idea:

Cualquier concepto $O1$ modelado en la ontología O , deberá coincidir con algún elemento $M1$ en un modelo relacional M .

3. Estrategia de solución al problema planteado

En el presente artículo, se desarrolló un procedimiento mediante el cual se clasifica la información de una base de datos, en conceptos, relaciones, roles y atributos. La figura 1 muestra la arquitectura del procedimiento propuesto.

La arquitectura mostrada en la figura 1, está compuesta de la manera siguiente:

1. **Fuente de datos:** una base de datos relacional en *PostgreSQL*⁵.
2. **Procedimiento para la clasificación de la información**, compuesto por dos métodos:
 - a) El *algoritmo de migración*. Este genera una base semántica construida de los elementos del esquema relacional y las relaciones entre estos. Estructurada a través de clases, propiedades, sus relaciones e instancias para conformar una ontología.
 - b) La *estrategia para la clasificación de la información*, compuesta por una serie de consultas *SPARQL* a través de las que el razonador evalúa la información de la BD.

⁵ Como caso particular en la presente investigación se utilizará *PostgreSQL*, pero el procedimiento es aplicable a cualquier gestor.

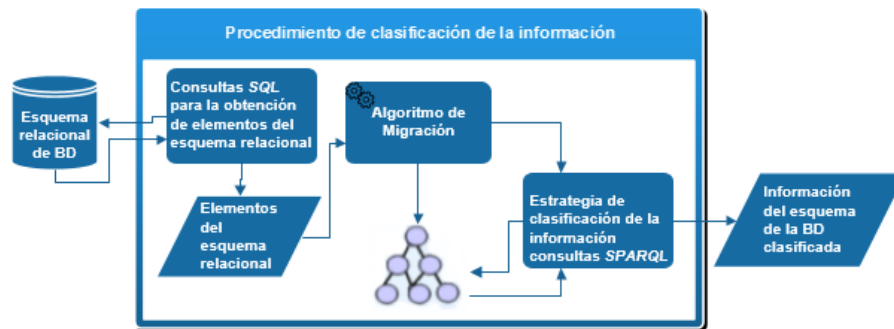


Figura 1. Arquitectura del procedimiento *BD-OntoClassification* propuesto como solución en este trabajo.

3.1. Algoritmo de migración

El algoritmo de migración propuesto, cumple con el principal requerimiento de estos sistemas: **la ontología que es construida a partir de una BD debe tener el más alto grado de similitud**⁶[11]. Este algoritmo está segmentado en dos pasos fundamentales: la extracción de elementos del esquema relacional de la base de datos y la generación de la ontología a partir de los elementos obtenidos.

■ Extracción de elementos del esquema de la BD

Los elementos del modelo relacional de la BD son obtenidos al consultar las tablas *constraint-column-usage*, *information-schema.columns* y *pg-catalog.pg-constraint*. Estas contienen la información de la composición del esquema del modelo, dígame: tablas, llaves primarias y foráneas, atributos y tuplas.

■ Generación de la ontología a partir de elementos del esquema

Una vez obtenidos los elementos del esquema relacional de la base de datos, a través de las tablas analizadas con anterioridad, se procede a la creación de la ontología.

Para explicar este proceso se tomará como ejemplo el esquema relacional de la base de datos **Control del proceso 1**⁷. Esta BD la componen siete entidades: **pieza, equipo, operación, obrero, auxiliar, directo y norma**, las que almacenan toda la información del proceso. Cada una de estas entidades presentan sus llaves, atributos y las relaciones entre cada una de estas. El esquema de la BD se muestra en la figura 2.

⁶ Esta similitud consiste en: la ontología resultante debe estar compuesta por los elementos que componen la BD utilizada como origen de datos.

⁷ Esta BD se diseñó para representar la información del proceso de fabricación de piezas en una industria mecánica

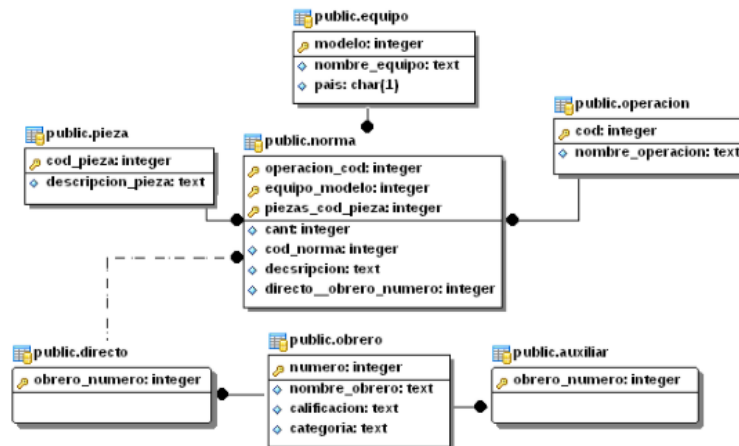


Figura 2. Esquema de la base de datos Control del proceso 1 realizada en este trabajo para ejemplificar el proceso.

Jerarquía de clases

Obtenidos los elementos del esquema relacional de la BD, se realiza un análisis para comprobar si existe repetición secuencial de algún atributo llave con la siguiente estructura: **(nombre de la tabla)-(atributo llave)**. Si existe esta repetición se está en presencia de una superclase y sus subclasses. Lo expuesto anteriormente se define a continuación:

Definición 1. Superclases y subclasses.

Sea A una entidad cualquiera en el modelo relacional, con un atributo tipo llave Apk . Si para dos entidades adicionales B y C , se tienen atributos llaves expresados de la forma $A-Bpk$ y $A-Cpk$, entonces B y C son especializaciones de la entidad A y serán expresadas como relaciones del tipo **Subclassof**.

En el caso de no presentar repetición en el campo de los atributos llaves, con la característica antes descrita, se toman las tablas que conforman el esquema de la BD y se crean un conjunto de clases. Estas clases se declaran de tipo *sibling* o clases independientes.

Propiedades de datos

A través de estas propiedades se describe la información del dominio de conocimiento modelado en la ontología. El procedimiento propuesto transforma los atributos de las tablas del esquema relacional como propiedades de datos en la forma siguiente:

1. Atributos de las tablas de la BD, son las propiedades de datos en la ontología.
2. Tablas de la BD se declaran como el dominio de cada propiedad de datos, pues de estas entidades es que se representa la información.
3. Los tipos de datos de cada atributo, estos se utilizan para declarar el rango de las propiedades de datos.

Para la declaración del **rango** de estas propiedades se realizó un proceso de conversión o *mapping*. Este proceso transformó los tipos de datos que son soportados en bases de datos a tipos de datos válidos en las ontologías. Algunos de los tipos de datos a los que se le realizó esta transformación se muestran en la tabla 1.

Cuadro 1. Muestra de tipos de datos soportados en BD mapeados a tipos de datos soportados por las ontologías.

Tipos de datos en la BD	Tipos de datos en la ontología
character	xsd:string
character varying	xsd:string
money	xsd:double
double precision	xsd:double

Relaciones del modelo ontológico

Para realizar este paso se tuvieron en cuenta los tipos de relaciones presentes en el diseño de bases de datos. A través de estas, se estructuraron las relaciones entre clases que componen el modelo ontológico.

- Relación (1:M): la clase que representa la entidad en la parte “mucho” de la relación, adquiere la llave primaria de la entidad “uno” como atributo y los atributos de la relación. Además la llave primaria de la entidad “uno” es adquirida como una propiedad de objeto funcional e inversa.
- Relación (1:1): si las dos entidades tienen sentido obligatorio, la llave primaria es adquirida por una de ellas como atributo de la otra. La clase que representa a la entidad que adquiere la llave, también adquiere a la otra entidad como propiedad de objeto para describir la relación.
- Relación (M:M): la clase que representa a la entidad surgida como resultado de la relación, adquiere las llaves primarias de las otras entidades para conformar el atributo llave de esta. Las entidades que intervienen en la relación son adquiridas como propiedades de objetos.

En el modelo ontológico las relaciones se representan de la siguiente forma:

Relación (M:M) \rightarrow Restricción de Mínima cardinalidad.

Relación (1:1) \rightarrow Restricción de Máxima cardinalidad.

Relación (1:M) \rightarrow Restricción de Mínima cardinalidad.

Creación de instancias de clases

Una vez obtenidas todas las tuplas de cada tabla en la base de datos, se declaran las instancias de clases en el modelo ontológico. Esta declaración se realiza sobre la siguiente definición:

Definición 2. Creación de instancias.

En una tabla T de un modelo relacional M , las instancias de T representada como clase en una ontología O , corresponderán al conjunto de tuplas que componen la tabla T en el modelo M .

Al declarar instancias de la clase pieza según la definición anterior, se nombra cada instancia según la clase (tabla) seguido del número correspondiente a dicha tupla. Las instancias de la clase pieza por ejemplo, son declaradas: *pieza1*, *pieza2*...*pieza-n*, de acuerdo con la cantidad de tuplas.

Luego de la declaración de las respectivas instancias de cada clase, se declaran las *data property assertion*, mediante las cuales se representan la información de cada instancia. La estructura de este axioma se realiza de la siguiente manera:

Instancia propiedad valor de la tupla

3.2. Estrategia para la clasificación de la información

El procedimiento que se propone se centra en el modelado de las definiciones de conceptos, relaciones, roles y atributos dados por Nicola Guarino en [4] como consultas *SPARQL*. Mediante las que se clasifica la información de la BD en las categorías antes señaladas. A continuación se describe el modelado de las relaciones, los atributos y roles en *SPARQL*.

Clasificación de las relaciones

Estas surgen al “relacionar dos conceptos o más conceptos cualesquiera, en un universo discursivo y como resultado surge un nuevo concepto” [4]. De acuerdo con lo anterior, el modelo de consulta que se diseñó para clasificar las relaciones cumple con los siguientes requisitos:

- Identificar las entidades que interviene en la relación.
- Identificar la forma mediante la cual se establece dicha relación.

En el siguiente cuadro se propone un modelo de consulta que cumple con los requisitos anteriormente descritos.

```
SELECT ?x ?y ?z
WHERE{
  ?x rdfs:subClassOf ?y.
  ?y owl:onProperty ?z.
}
```

La tripla **?x rdfs:subClassOf ?y**, permite obtener las formas de relación⁸ presente en la entidad correspondiente a la variable (**x**), representadas a través del axioma *subClassOf*. Mientras que las formas de relación corresponden a la variable (**y**).

Por otra parte, la tripla **?y owl:onProperty ?z**, permite identificar las entidades que intervienen en la forma de relación representada en la variable (**y**) como propiedades de datos. Las entidades transformadas en propiedades se obtienen en la variable (**z**).

⁸ Como caso particular de la presente investigación las formas de relación son de cardinalidad

Clasificación de atributos

Guarino reconoce a un atributo como, “un concepto que, en el dominio de interés, tiene una interpretación” [4]. Con relación a lo anterior, se entiende como atributo, todo concepto que a través de su interpretación se obtenga información de una entidad en el dominio de interés.

Ontológicamente hablando, para clasificar un atributo se hace indispensable obtener:

- El dominio que se describe: la clase.
- El concepto mediante el cual se describe dicho dominio: la propiedad de dato.

A continuación se muestra un modelo de consulta *SPARQL*, mediante el cual se definen los atributos del modelo.

```
SELECT ?x ?y
WHERE{
?x rdfs:domain ?y.
?x rdf:type owl:DatatypeProperty.
}
```

En la tripla **?x rdfs:domain ?y**, los conceptos correspondientes al valor de la variable (**y**) representan los dominios de los cuales se representa la información, representados a través de los conceptos obtenidos en la variable (**x**).

Por otro lado, la tripla **?x rdf:type owl:DatatypeProperty**, garantiza que los conceptos de la variable (**x**), sean del tipo propiedad de dato.

Clasificación de roles

Un rol es, “un concepto que participa de una relación particular, no semánticamente rígido⁹, si al participar de esta relación, es necesario una Entidad para explicarlo” [4].

De acuerdo con la anterior definición, las entidades dentro del modelo que cumplen la condición para ser clasificada como un rol son las subclases. Debido a que para definir dichas entidades en el dominio de conocimiento será necesario utilizar la superclase.

El modelo de consulta que se propone a continuación, permite obtener los roles presentes en el modelo.

```
SELECT ?x ?y
WHERE{
?x rdfs:subClassOf ?y.
?y rdf:type owl:Class.
}
```

En la tripla **?x rdfs:subClassOf ?y**, el valor de la variable (**x**) corresponde a las subclases del modelo y el valor de la variable (**y**) a la superclase. Esta

⁹ No se puede realizar definición alguna de la entidad a través de la relación que participa.

puede ser una clase del dominio o la clase *owl:Thing* como superclase general en el modelo.

Para garantizar que la superclase sea una de las clases del dominio se ha añadido la tripla `?y rdf:type owl:Class`, la que garantiza que la variable (**y**) sea una clase en el modelo, o sea, que los conceptos correspondientes al valor de la variable (**x**) estén fundados en (**y**).

4. Principales Resultados

Para evaluar el procedimiento propuesto se realizaron diez (10) casos experimentales¹⁰. La tabla 2 muestra la composición de estas bases de datos¹¹.

Mientras que la figura 3, muestra la ontología obtenida a través del algoritmo de migración de la BD **Control del proceso 1**, que corresponde al primer caso experimental. Esta ontología presenta la misma cantidad de elementos de la fuente de datos con una expresividad *ALIN(D)*, dado a que todo modelo ontológico obtenido a partir de una base de datos está compuesto (en términos de la expresividad del modelo) por:

- Propiedades Inversas(*I*).
- Restricciones de Cardinalidad(*N*).
- Propiedades de tipos de datos y tipos de datos(*D*).

Cuadro 2. Bases de datos relacionales utilizadas en los casos experimentales.

Nombre	Descripción del dominio	NT	NA	NPK	NFK
Control del Proceso 1	Producción de piezas	7	18	7	4
<i>Air Reservation</i>	Reservaciones de vuelos	10	39	10	7
BMEWS	Inteligencia de negocio	6	8	6	2
<i>Children Institution</i>	Casas de acogida de niños (Unicef)	9	10	9	2
<i>Children of Family Support</i>	Adopción de niños (Unicef)	10	13	10	7
<i>Gaming Web Sites</i>	Video juegos	10	32	10	11
ISWC	Publicaciones científicas	9	33	9	8
<i>National Healt</i>	Control de pacientes	7	25	7	6
<i>Open University</i>	Maticulas universitarias	10	24	10	10
<i>Theater Booking</i>	Reservación de teatros	8	29	8	6

¹⁰ Cada caso experimental cuenta como origen de datos un modelo relacional extraído de *Data Base Answers*(www.databseanswer.org/data-model)

¹¹ Significados de las abreviaturas en la tabla: NT= número de tablas, NA= número de atributos, NPK= número de llaves primarias y NFK= número de llaves fráneas.

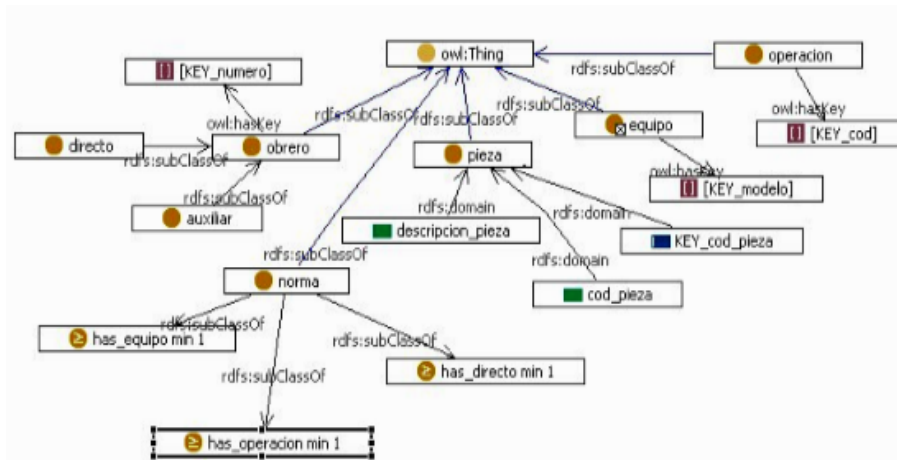


Figura 3. Ontología resultante una vez aplicado el algoritmo de migración a la BD Control del Proceso 1.

4.1. Evaluación de la Eficacia del algoritmo de migración

La **Eficacia** del algoritmo de migración se determinó a través de la media ponderada de las **Medidas F** de las correspondencias entre conceptos (C), atributos (A) y relaciones (R). Esta métrica determina la calidad de un sistema en función de la cantidad de recursos que requiere para realizar el objetivo para el que se diseñó.[11]

$$Ef(Caso) = \frac{C}{C + A + R} * F_{concep.} + \frac{A}{C + A + R} * F_{atrib.} + \frac{R}{C + A + R} * F_{relac.}$$

La figura 4¹² muestra una gráfica que ilustra los valores de **Eficacia** para cada caso experimental realizado para probar el algoritmo de migración.

Al determinar la media muestral de los valores de eficacia para cada caso experimental se obtiene una $\overline{M}(Ef) = 0.98$. Con este valor se determina que el algoritmo de migración propuesto en el presente artículo cumple con el principal requerimiento para estos sistemas, **el máximo grado de similitud entre el modelo resultante y el modelo utilizado como origen de datos.**

4.2. Resultados obtenidos con el procedimiento de clasificación de la información

El procedimiento de clasificación de la información que se propuso en la sección 3.2, evalúa como conceptos, atributos, relaciones y roles la información

¹² Significado de las abreviatutas en la figura: CP= Control de Proceso 1, AR= Air Reservation, BWS= BMEWS, ChI= Children Institution, ChF= Children of Family Support, GWS= Gaming Web Sites, ISW= ISWC, NH= National Health, OU= Open University, TB= Theater Booking

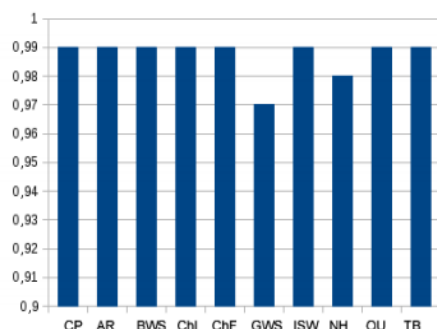


Figura 4. Valores de Eficacia por cada caso experimental.

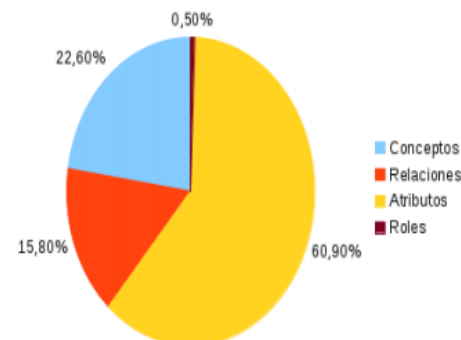


Figura 5. Porcientos que representan cada elemento clasificado en la composición de los elementos de una base de datos.

contenida en la base de datos. En la figura 5, se presentan los porcentajes de los elementos que son clasificados para los casos experimentales realizados.

Como se aprecia en la figura 5, los atributos ocupan el más alto porcentaje, pues estos **representan la información contenida en las bases de datos**. No es así el caso de los roles que representan el menor porcentaje debido a que no todas las bases de datos existentes requieren en su diseño de especializaciones entre entidades para representar la información. Esto depende del dominio para el cual se diseñe la BD.

4.3. Comparación del procedimiento propuesto y los sistemas existentes

A continuación se realiza una breve comparación entre los sistemas de migración y/o *upgrade* analizados en la sección 2 y el procedimiento propuesto en el presente artículo. Para esta comparación se tendrán en cuenta cuatro aspectos fundamentales: el lenguaje de representación del modelo resultante(LR), proceso de migración y/o *upgrade*(M y/o U), el establecimiento de correspondencias y la posibilidad de clasificación de información. La tabla 3, muestra los resultados generales tras comparar el procedimiento *BD-OntoClassification* y los sistemas existentes.

Como se aprecia en la tabla 3, en torno al aspecto de migración y *upgrade*, el procedimiento propuesto presenta resultados similares y bien puede incluirse dentro del grupo de estos sistemas. En cuanto al establecimiento de correspondencias semánticas, el procedimiento del presente trabajo permite definir correspondencias entre las BD y la ontología, al estar la ontología resultante conformada con los elementos que componen el esquema de la BD relacional.

Cuadro 3. Comparación del procedimiento propuesto y los sistemas existentes.

Sistemas	LR	M y/o U	Corresp.	CI
<i>OntoLift</i>	<i>RDF</i>	x	x	-
<i>Kaon-Reverse</i>	<i>RDF/DAML + OIL</i>	x	x	-
<i>OntoStudio</i>	<i>RDF</i>	x	x	-
<i>D2R-MAP</i>	<i>D2R</i>	x	x	-
<i>Ontology Based Q. P.</i>	<i>RDF</i>	x	x	-
<i>BD-OntoClassification</i>	<i>RDF/OWL</i>	x	x	x

Por otra parte, referente a la clasificación de la información, los sistemas analizados no presentan ningún mecanismo para realizar este proceso, mientras que el procedimiento propuesto presenta una estrategia de razonamiento a través de consultas *SPARQL*, que permite a un razonador determinar conceptos, relaciones, roles y atributos en la información almacenada en bases de datos.

5. Conclusiones y referencias para futuros trabajos

- El algoritmo de migración propuesto probado en diez bases de datos relacionales, muestra buenos resultados al proporcionar el máximo grado de similitud entre las bases de datos y la ontología resultante, con un alto grado de eficacia.
- La estrategia de clasificación propuesta, identifica conceptos, relaciones, atributos y roles en la información almacenada en una base de datos, en total correspondencia con las definiciones de estos elementos sugeridas por Nicola Guarino.
- El mayor porcentaje de la composición de las bases de datos corresponde a los atributos, siendo estos los elementos que permiten la representación de la información, el principal objetivo de las bases de datos.

Referencias

1. Gary W. Hansen, J.V.H.: Sistemas de Bases de Datos. (1995)
2. Blanco, J.I., Martínez, C., Amparo, V.M.: Arquitectura para la integración de esquemas relacionales difusos basada en ontologías: una aplicación para la web. Dpto. Ciencias de la Computación e I. A., Universidad de Granada. (2006)
3. Berners-Lee, T.: The semantic web. (2001)
4. Guarino, N.: Concepts, attributes, and arbitrary relations: Some linguistic and ontological criteria for structuring knowledge bases. (1992)
5. Berners-Lee, T.: Relational databases on the semantic web. (1998)
6. S, V.R.O.D.S., R., S.: Ontolift prototype. WonderWeb Deliverable D11 (2003)
7. Stojanovic L, S.N., R, V.: A reverse engineering approach for migrating data intensive web sites to the semantic web. IFIP WCC2002 (2002)
8. Volz Raphael, Handschuh Siegfried, S.S.S.L., Nenad, S.: Unveiling the hidden bride: Deep annotation for mapping and migrating legacy data to the semantic web. Journal on Web Semantics (2004)

9. Christian, B.: D2r map a database to rdf mapping language. In 12th Intl World Wide Web Conference Budapest May 2003 (2003)
10. Chokrid, B.N., Johann, C.F.: Ontology based query processing in database management system. ODBASE - OTM 2003 (2003)
11. Barrasa, R.J.: Modelo para la definición automática de correspondencias semánticas entre ontologas y modelos relacionales. PhD thesis, Universidad Politécnica de Madrid (2007)