

## Estudio y formalización del proceso de mezcla de ontologías

Mariano Fernández-López<sup>(1)(2)</sup>, Asunción Gómez-Pérez<sup>(1)</sup>,  
José Ángel Ramos<sup>(1)</sup>

(1) OEG - Facultad de Informática. Universidad Politécnica de Madrid (UPM)  
Campus de Montegancedo, s/n.  
28660 Boadilla del Monte. Madrid. Spain  
{mfernandez,asun,jarg}@fi.upm.es

(2) Escuela Politécnica Superior. Universidad San Pablo CEU  
Ctra. de Boadilla del Monte km 5,300  
28668 Boadilla del Monte. Madrid. Spain  
mfernandez.eps@ceu.es

### Resumen

Desde hace tiempo la mezcla de ontologías es una actividad necesaria, sin embargo, los actuales métodos de mezcla de ontologías no tienen una casuística detallada ni una formalización precisa. Para la validación de estos métodos, es conveniente disponer de una casuística lo más completa posible. Por ello, en este artículo se presenta el modelo OEGMerge, desarrollado a partir de la experiencia del Grupo de Ingeniería Ontológica (OEG) de la UPM, en el que se describe detallada y formalmente la casuística de mezcla y las acciones a realizar en cada caso. En esta primera aproximación sólo se abarca taxonomía de conceptos, atributos y relaciones.

**Palabras clave:** Ontología, Mezcla, Modelo.

### 1. Introducción

Las ontologías permiten capturar conocimiento consensuado de un dominio dado de una forma genérica y formal, para ser reutilizado y compartido por aplicaciones y grupos de personas. Desde esta definición se puede inferir erróneamente que hay una sola ontología para modelar un dominio. Sin embargo, se puede encontrar en la literatura varias ontologías que modelan el mismo conocimiento de diferentes formas. Por ejemplo, hay varias ontologías *top-level* que difieren en el criterio seguido para clasificar los conceptos más generales de la taxonomía (ej. [7]), y en el campo del comercio electrónico hay varios estándares e

iniciativas conjuntas para la clasificación de productos y servicios (UNSPSC<sup>1</sup>, e-cl@ss<sup>2</sup>, RosettaNet<sup>3</sup>, etc.). Esta heterogeneidad en ontologías también se presenta en otros muchos dominios (medicina, ley, arte, ciencias, etc.).

Para Noy y Musen [10] el alineamiento de ontologías (*ontology alignment*) consiste en el establecimiento de diferentes clases de *mappings* (o correspondencias) entre los componentes de dos ontologías, preservando con esto las ontologías originales; y la mezcla de ontologías (*ontology merging*) propone generar una única ontología a

<sup>1</sup> <http://www.unspsc.org/>

<sup>2</sup> <http://www.eclass.de/>

<sup>3</sup> <http://www.rosettanel.org/>

partir de las ontologías originales. Este artículo se va a centrar en la mezcla de ontologías, aunque algunas ideas presentadas puedan ser utilizadas para el alineamiento de las mismas. De hecho, antes de mezclar dos ontologías hay que establecer correspondencias entre los términos de las ontologías a mezclar.

Hasta ahora, aunque se han publicado algoritmos que permiten mezclar ontologías sencillas, los autores de estos algoritmos no han proporcionado las casuísticas que se pueden presentar a la hora de hacer mezcla, ni tan siquiera de taxonomías.

En el Grupo de Ingeniería Ontológica (OEG<sup>4</sup>) se ha elaborado, e implementado como un servicio de WebODE [1], el modelo **OEGMerge**, para la mezcla de ontologías que especifica la casuística encontrada en las mezclas realizadas en diferentes proyectos. Una de las características más relevantes de este modelo es que no se trata de un algoritmo, sino de un sistema basado en reglas. De hecho, las características del problema de mezcla (uso de conocimiento incompleto, de heurísticas, de conocimiento de ingenieros ontológicos, etc.) hacen pensar que un modelo basado en reglas es más adecuado que un algoritmo procedimental, teniendo como base de conocimiento las ontologías y el conjunto de *mappings* entre ellas.

En la sección 2 de este artículo se revisan los métodos de mezcla de ontologías propuestos; en la sección 3 se presenta el conocimiento previo del que se parte, formalizado de manera sencilla; en la sección 4 se presenta el modelo **OEGMerge**; en la sección 5 se demuestra la completud del modelo; en la sección 6 se presenta la implementación parcial de la mezcla definida en OEGMerge: ODEMerge; y en la sección 7 se muestran las conclusiones y los trabajos futuros.

## 2. Estado de la cuestión sobre los métodos de mezcla de ontologías

Las contribuciones más importantes sobre cómo mezclar ontologías son ONIONS [6], FCA-Merge [14] y PROMPT [11].

**ONIONS**. Con este método se puede crear una biblioteca de ontologías provenientes de diferentes fuentes. En los primeros estadios de la integración, las fuentes son unificadas sin considerar las teorías generales que se usan en la modelización (teorías de parte-todo o desconectividad, por ejemplo).

<sup>4</sup> <http://www.oeg-upm.net>

Después, estas teorías se incorporan como una base de la biblioteca y se integran como caras de un mismo poliedro. De esta manera, las ontologías de la biblioteca están conectadas a través de los términos de las teorías genéricas compartidas. Desde el punto de vista estricto no se puede considerar mezcla de ontologías puesto que no se genera una ontología única a partir de las fuentes, pero sí se trata la biblioteca completa como si fuera una ontología.

**FCA-Merge** [14]. Este enfoque es muy diferente al de los otros presentados en esta sección. FCA-Merge toma como entrada dos ontologías a mezclar y un conjunto de documentos del dominio de las ontologías. La mezcla se hace identificando los documentos que contienen información que se identifica como instancias que pertenecen a conceptos de ambas ontologías.

El método **PROMPT** [11]. La principal asunción de PROMPT es que las ontologías a mezclar están formalizadas con un modelo de conocimiento común basado en marcos. Existe un *plug-in* de Protégé-2000 que mezcla ontologías de acuerdo al método PROMPT. Este método propone primero elaborar una lista de las operaciones a ser realizadas para mezclar las dos ontologías (por ejemplo: mezclar dos clases, mezclar dos atributos, etc.). Esta actividad es llevada a cabo automáticamente por el *plug-in* de PROMPT. Seguidamente comienza un proceso cíclico supervisado por un ingeniero ontológico. En cada ciclo, se debe seleccionar y ejecutar una operación de la lista de las posibles operaciones a realizar, incluidas las operaciones para evitar conflictos.

Las propuestas anteriormente mencionadas suponen un punto de inicio en la investigación sobre la mezcla de ontologías, sin embargo, sus autores no proporcionan detalles de las casuísticas que se presentan a la hora de llevar a cabo la mezcla. A lo largo de los apartados siguientes, se va a presentar OEGMerge, un modelo, implementado parcialmente en WebODE [1], que está detallado utilizando una serie de reglas basadas en los casos que se pueden presentar al mezclar taxonomías. Las premisas de estas reglas, serán los casos posibles a los que la mezcla se enfrenta.

Existen otros sistemas de mezcla de ontologías, pero que no representan métodos en sí, sino soluciones para un determinado lenguaje o para una determinada acción: CUPID [9], OntoMerge<sup>5</sup>, GLUE [4], etc.

<sup>5</sup> <http://cs-www.cs.yale.edu/homes/dvm/daml/ontology-translation.html>

### 3. Conocimiento previo

Para mostrar el modelo, primero se proporcionarán las definiciones de ontología y de conjunto de mappings en que se apoya. Así, esto funcionará como base de conocimiento del sistema basado en reglas.

La formalización de ontología que se expresa a continuación se basa en la definición de Studer y colegas [13]: una ontología es una especificación formal y explícita de una conceptualización compartida.

**Definición 1.-** Una ontología es una tupla  $O = (C, A, atributoDe_O, subclaseDe_O, R)$  donde  $C$  es un conjunto de conceptos,  $A$  es un conjunto de atributos,  $atributoDe_O \subseteq A \times C$ ,  $subclaseDe_O \subseteq C \times C$  y  $R \subseteq C \times C$  el conjunto de relaciones ad hoc.

Entre las observaciones a tener en cuenta sobre la definición 1, en primer lugar, hay que hacer notar que está formulada para cubrir las necesidades de la primera versión de OEGMerge. Para otros propósitos, sería necesaria una definición más amplia que incluyera otros componentes de las ontologías. En segundo lugar, las relaciones *atributoDe<sub>O</sub>* y *subclaseDe<sub>O</sub>* llevan la etiqueta O para indicar que está particularizadas a la ontología O. Por último, los términos “concepto”, “atributo” y “relación” se consideran primitivos a efectos de este artículo y, por tanto, no se van a definir formalmente aquí.



Figura 1: Ejemplo de ontología de animales

Por ejemplo, la ontología de la figura 1 se representa, de acuerdo con la definición 1, de la siguiente manera:

$$\begin{aligned}
 O_{animales} = & (C_{animales} = \{hombre, perro, animal\}, \\
 & A_{animales} = \{fecha\ de\ nacimiento, \\
 & \quad \quad \quad lugar\ de\ nacimiento\}, \\
 & atributoDeO_{animales} = \\
 & \quad \{(fecha\ de\ nacimiento, animal), \\
 & \quad \quad (fecha\ de\ nacimiento, perro), \\
 & \quad \quad (fecha\ de\ nacimiento, hombre), \\
 & \quad \quad (lugar\ de\ nacimiento, animal), \\
 & \quad \quad (lugar\ de\ nacimiento, perro), \\
 & \quad \quad (lugar\ de\ nacimiento, hombre)\}, \\
 & subclaseDeO_{animales} = \\
 & \quad \{(hombre, animal), (perro, animal)\}, \\
 & R_{animales} = \emptyset)
 \end{aligned}$$

Nótese que la herencia de atributos en la ontología se ha reflejado en este caso y completamente en el conjunto de atributos de la ontología.

**Definición 2.-** El conjunto de mappings entre las ontologías  $O_1$  y  $O_2$  es una tupla  $M = (O_1, O_2, elMismoQue_M, subclaseDe_M, superclaseDe_M)$  donde  $elMismoQue_M \subseteq \mathcal{P}(T_1) \times \mathcal{P}(T_2)$  y  $subclaseDe_M, superclaseDe_M \subseteq C_1 \times C_2$ , siendo  $C_1$  el conjunto de conceptos de  $O_1$ , y  $C_2$  el conjunto de conceptos de  $O_2$ , siendo  $\mathcal{P}$  todos los subconjuntos posibles de un conjunto, y siendo  $T_1 = C_1 \cup A_1 \cup R_1$  y  $T_2 = C_2 \cup A_2 \cup R_2$  los términos de  $O_1$  y  $O_2$ .

Obsérvese que la relación *elMismoQue<sub>M</sub>* se establece entre cualquier término, por ejemplo, entre un atributo y una relación, o incluso entre conjuntos de términos (por eso se hace referencia a sus partes,  $\mathcal{P}$ ). Así, por ejemplo, un atributo de una ontología, como es el caso de *nombre*, puede corresponderse con varios atributos de la otra ontología: *nombre*, *apellido 1* y *apellido 2*.

A modo de ejemplo, el conjunto de los mappings de la figura 2 se representa, de acuerdo con la definición 2, de la siguiente manera:

$$\begin{aligned}
 M_{animales} = & (O_1, O_2, elMismoQueM_{animales} = \\
 & \quad \{(animal, animal), \\
 & \quad \quad (hombre, animal\ racional)\}, \\
 & subclaseDeM_{animales} = \emptyset, \\
 & superclaseDeM_{animales} = \\
 & \quad \{(perro, San\ Bernardo), \\
 & \quad \quad (perro, Pastor\ alemán)\})
 \end{aligned}$$

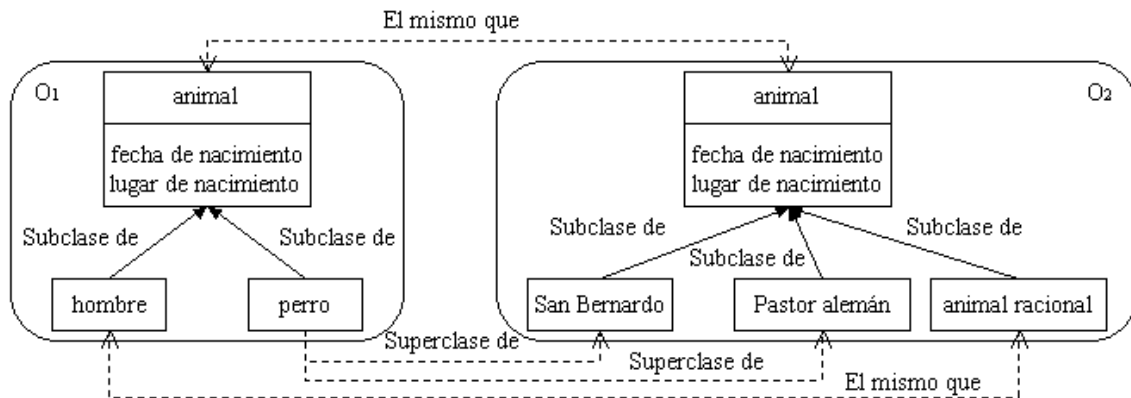


Figura 2: Ejemplo de *mappings* entre dos ontologías

Al igual que ocurría con la definición 1, aquí también se están considerando sólo los componentes para la primera versión de OEGMerge. Además, según la definición 2, un *mapping* establece relaciones de identidad, de *superclase de* y de *subclase de* entre los conceptos de la ontología  $O_1$  y de la ontología  $O_2$ . En futuros trabajos, se extenderá la semántica para abarcar etiquetas como *unión*, *parte de*, etc. Por último, la generación automática de estos *mappings*, utilizando similitudes entre etiquetas, documentos, estructuras, instancias, etc. está fuera del alcance de este artículo (véanse, por ejemplo, [2] o [5]).

Una vez que se ha definido qué es una ontología y el conjunto de *mappings* entre las ontologías, se está en condiciones de definir el modelo de mezcla OEGMerge.

#### 4. OEGMerge

El modelo OEGMerge surge de la experiencia de mezcla de ontologías en proyectos del Grupo OEG. Esta experiencia realizada metódicamente ha dado lugar a la identificación de una casuística, detallada en este artículo. El modelo parte de dos ontologías originarias para producir una nueva ontología que contenga todo el conocimiento de las dos originarias.

Es **importante** tener en cuenta que el modelo aquí expuesto no considera ontologías con contradicciones, por ejemplo, que en una se modelice que  $a$  es subclase de  $b$ , y en otra que  $b$  es subclase de  $a$ . Igualmente, no se consideran ontologías con redundancias de modo que no pueda ocurrir que un componente de una ontología sea

similar a más de un componente de la otra. Por esta razón, se recomienda llevar a cabo una tarea de evaluación antes de realizar la mezcla.

Las dos ontologías originales se van a identificar a partir de ahora como ontología base y ontología auxiliar. La elección de la ontología base y, por tanto, de la auxiliar no influye en el resultado final de la mezcla. Se realiza la diferencia sólo para tomar una de las ontologías como punto de referencia. No obstante, la experiencia recomienda elegir como ontología base la más completa por número de conceptos y atributos.

**Definición 3.-** La *mezcla* de dos ontologías  $O_{aux}$ ,  $O_{base}$  utilizando un conjunto de *mappings*  $M$  es una función *MezclaOntología*:  $O_{aux} \times O_{base} \times M \rightarrow O_{res}$  que obtiene la ontología resultado después de aplicar los siguientes pasos:

- Paso 1) Se establece  $O_{res} = O_{base}$
- Paso 2) Se ejecutan las reglas que se muestran en la Tabla 1, que, a su vez, puede ejecutar la acción *MezclaIdentidad*. Esta acción también ejecuta una serie de reglas: las mostradas en la Tabla 2.

Las condiciones de la casuística de las tablas 1 y 2 se muestran en la Tabla 3, y las acciones, en la Tabla 4.

A la hora de establecer las acciones a realizar en cada caso de los identificados en las tablas hemos seguido las siguientes heurísticas:

- a) *Preferencia de las primitivas de representación que sean más expresivas.* Así, por ejemplo, se prefiere un concepto a un tipo básico de un atributo.
- b) *Resolución de carencias.* Aunque un conocimiento esté expresado en una ontología, pero no en la otra, debe aparecer en la ontología

resultado. Por ejemplo, si el concepto C está representado en alguna ontología, aunque no aparezca en la otra, debe estar presente en la ontología resultado.

Puesto que quizás no todos los investigadores estén de acuerdo con las tablas resultantes, pueden hacer

los cambios oportunos en el procedimiento de mezcla, modificando la acción a realizar en cada caso. Aun así, los casos que se consideran aquí serían los mismos, sólo implicando cambio en su tratamiento.

**Tabla 1: Reglas de definición de la función MezclaOntología**

Caso	Condición	Acción
1	$\neg \text{elMismoQue}_M(c_{aux}, c_{base}) \wedge$ $(\text{subclaseDe}_M(c_{base}, c_{aux}) \vee \text{superclaseDe}_M(c_{aux}, c_{base}))$	$\text{insertarConcepto}(c_{aux}, o_{res}) \wedge$ $\text{insertarSubclaseDe}(c_{base}, c_{aux})$
2	$\neg \text{elMismoQue}_M(c_{base}, c_{aux}) \wedge$ $(\text{subclaseDe}_M(c_{aux}, c_{base}) \vee \text{superclaseDe}_M(c_{base}, c_{aux}))$	$\text{insertarSubclaseDe}(c_{aux}, c_{base})$
3	$\neg \text{elMismoQue}_M(c_{aux}, c_{base}) \wedge$ $\neg \text{subclaseDe}_M(c_{aux}, c_{base}) \wedge \neg \text{superclaseDe}_M(c_{aux}, c_{base})$	$\text{insertarConcepto}(c_{aux}, o_{res})$
4	$\neg \text{elMismoQue}_M(c_{base}, c_{aux}) \wedge$ $\neg \text{subclaseDe}_M(c_{base}, c_{aux}) \wedge \neg \text{superclaseDe}_M(c_{base}, c_{aux})$	<<Sin acción>>
5	$\text{elMismoQue}_M(c_{aux}, c_{base})$	$\text{MezclaIdentidad}(c_{aux}, c_{base}, o_{aux}, o_{base}, \text{mapping})$

**Tabla 2: Reglas de definición de la función MezclaIdentidad**

Caso	Condición	Acción
S-1	$\text{atributoSimilar}(a_{aux}, a_{base})$	<<Sin acción>>
S-2	$\neg \text{atributoSimilar}(a_{base}, a_{aux}) \wedge$ $\neg \text{atributoSimilarRelacion}(a_{base}, r_{aux}) \wedge$ $\neg \text{atributoSimilarConcepto}(a_{base}, c_{aux})$	<<Sin acción>>
S-3	$\neg \text{atributoSimilar}(a_{aux}, a_{base}) \wedge$ $\neg \text{atributoSimilarRelacion}(a_{aux}, r_{base}) \wedge$ $\neg \text{atributoSimilarConcepto}(a_{aux}, c_{base})$	$\text{insertarAtributo}(a_{aux}, c_{res}, o_{res})$
S-4	$\text{relacionSimilar}(r_{aux}, r_{base})$	<<Sin acción>>
S-5	$\neg \text{relacionSimilar}(r_{base}, r_{aux}) \wedge$ $\neg \text{relacionSimilarConceptoDosRelaciones}(r_{base}, c_{aux}, r1_{aux}, r2_{aux})$	<<Sin acción>>
S-6	$\neg \text{relacionSimilar}(r_{aux}, r_{base}) \wedge$ $\neg \text{relacionSimilarConceptoDosRelaciones}(r_{aux}, c_{base}, r1_{base}, r2_{base})$	$\text{insertarRelacion}(r_{aux}, o_{res})$
S-7	$\text{relacionSimilarConceptoDosRelaciones}(r_{aux}, c_{base}, r1_{base}, r2_{base})$	<<Sin acción>>
S-8	$\text{relacionSimilarConceptoDosRelaciones}(r_{base}, c_{aux}, r1_{aux}, r2_{aux})$	$\text{insertarRelacion}(r1_{aux}, o_{res}) \wedge$ $\text{insertarRelacion}(r2_{aux}, o_{res}) \wedge$ $\text{insertarConcepto}(c_{aux}, o_{res}) \wedge$ $\text{borrarRelacion}(r_{base}, o_{res})$
S-9	$\text{atributoSimilarConcepto}(a_{aux}, c_{base})$	$\text{insertarRelacionTiene}(c_{aux}, c_{base}, o_{res})$
S-10	$\text{atributoSimilarConcepto}(a_{base}, c_{aux})$	$\text{insertarConcepto}(c_{aux}, o_{res}) \wedge$ $\text{borrarAtributo}(a_{base}, c_{base}, o_{res}) \wedge$ $\text{insertarRelacionHas}(c_{aux}, c_{base}, o_{res})$
S-11	$\text{atributoSimilarRelacion}(a_{base}, r_{aux})$	$\text{insertarRelacion}(r_{aux}, o_{res})$
S-12	$\text{atributoSimilarRelacion}(a_{aux}, r_{base})$	<<Sin acción>>
S-13	$\text{elMismoQue}_M(\text{padre}(c_{base}), \text{padre}(c_{aux}))$	<<Sin acción>>
S-14	$\neg \text{elMismoQue}_M(\text{padre}(c_{base}), \text{padre}(c_{aux}))$	$\text{insertarSubclaseDe}(c_{base}, \text{padre}(c_{aux}), o_{res})$
S-15	$\text{elMismoQue}_M(\text{padre}(c_{aux}), \text{hermano}(c_{base}))$	$\text{insertarSubclaseDe}(c_{base}, \text{hermano}(c_{base}), o_{res})$ $\wedge \text{borrarSubclaseDe}(c_{base}, \text{padre}(c_{base}), o_{res})$
S-16	$\text{elMismoQue}_M(\text{padre}(c_{base}), \text{hermano}(c_{aux}))$	<<Sin acción>>

En la columna de acción de la Tabla 1 y de la Tabla 2 aparece en nueve ocasiones “<<Sin Acción>>”. En la mezcla, según los criterios seguidos, no ha de realizarse ninguna acción para estos casos. Sin embargo, puesto que lo que se pretende es detallar toda la casuística que la mezcla va a encontrar, hemos reflejado el caso aunque no conllevara la realización de una acción.

En la Tabla 1, la regla 1 establece que cuando hay un concepto auxiliar no se corresponde con uno base y es padre de uno base (por mapping *subclase de* o *superclase de*), se debe insertar el concepto auxiliar y su relación taxonómica en la ontología resultado. La regla 2 establece que cuando un concepto base que no se corresponde con un auxiliar y es padre de uno auxiliar (por mapping *subclase de* o *superclase de*), se debe insertar la relación taxonómica entre los dos en la ontología resultado. La regla 3 establece que si un concepto auxiliar no está relacionado de ninguna forma con ningún concepto base (según el conjunto de *mapping*), se ha de insertar en la ontología resultado. La regla 4 establece que si un concepto base no está relacionado de ninguna forma con ningún concepto auxiliar (según el conjunto de *mapping*), no se debe realizar ninguna acción. La regla 5 establece que si el concepto de la ontología auxiliar es el mismo, según el conjunto de *mapping*, que el de la ontología base, entonces se lleva a cabo la acción *MezclaIdentidad*, definida en la Tabla 2. Aquí es importante tener en cuenta que, en caso de estar trabajando con una herramienta gráfica, hay que borrar las relaciones taxonómicas redundantes que surjan después de introducir nuevos enlaces *subclase de*. Así, por ejemplo, si se tiene *subclaseDe(a, b)*, *subclaseDe(b, c)* y *subclaseDe(a, c)*, en la herramienta gráfica habría que borrar *subclaseDe(a, c)*.

En la Figura 2, tomando como ontología base  $O_1$  y como ontología auxiliar  $O_2$ , se identifican, por

ejemplo, un caso 2 entre *perro* y *San Bernardo*, y un caso 5 entre *hombre* y *animal racional*.

El primer caso de la Tabla 2 (S-1) se corresponde con la situación en que un atributo del concepto base es similar con otro del concepto auxiliar, lo que no provoca acción de mezcla ninguna. Si el atributo auxiliar no existe ni en la ontología base ni como atributo, ni como relación, ni como concepto (caso S-2), no se llevará a cabo ninguna acción de mezcla. Si el atributo auxiliar no existe ni en la ontología base ni como atributo (caso S-3), ni como relación, ni como concepto, se insertará el atributo auxiliar en el concepto resultado.

Los siguientes cinco casos corresponden a las situaciones que se dan en la mezcla de relaciones: existencia de una relación auxiliar y base similares (S-4), no similitud entre relación base y relación auxiliar o conjunto auxiliar de concepto y relaciones (S-5), no similitud entre relación auxiliar y relación base o conjunto base de concepto y relaciones (S-6), existencia de similitud entre relación auxiliar y conjunto de concepto y relaciones base (S-7) y existencia de similitud entre relación base y conjunto de concepto y relaciones auxiliares (S-8).

Los dos casos siguientes se corresponden con la similitud entre atributo y concepto, en sus dos combinaciones: atributo auxiliar y concepto base (S-9) y atributo base y concepto auxiliar (S-10).

Los dos casos siguientes se corresponden con la similitud entre atributo y relación, en sus dos combinaciones: atributo auxiliar y relación base (S-11) y atributo base y relación auxiliar (S-12).

En el ejemplo de la Figura 2, tomando como ontología base  $O_1$  y como ontología auxiliar  $O_2$ , se puede identificar un caso S-13 entre *perro* y *animal racional*, donde sus respectivos padres (*animal* y *animal*) son similares también.

**Tabla 3: Predicados de las condiciones**

Predicado	Definición formal
atributoSimilar( $a_1, a_2$ )	elMismoQue $_M(a_1, a_2)$
atributoSimilarConcepto( $a_1, c_2$ )	elMismoQue $_M(a_1, c_2)$
atributoSimilarRelacion( $a_1, r_2$ )	elMismoQue $_M(a_1, r_2)$
relaciónSimilar( $r_1, r_2$ )	elMismoQue $_M(r_1, r_2)$
relacionSimilarConceptoDosRelaciones( $r_1, c_2, r_{21}, r_{22}$ )	elMismoQue $_M(r_1, c_2) \wedge$ $\exists c_i, c_f \in C_1, C_2$ $\text{tq } \{r_1(c_i, c_f)\} \in R_1 \wedge \{r_{21}(c_i, c_2), r_{22}(c_2, c_f)\} \in R_2$
padre( $c_1$ )	$\exists c_1' \in C_A \text{ tq } (c_1, c_1') \in \text{subclaseDe}O_A$
hermano( $c_{11}$ )	$\exists c_1', c_{12} \in C_A \text{ tq } \{(c_{11}, c_1'), (c_{12}, c_1')\} \in \text{subclaseDe}O_A$

Tabla 4: Predicados de las acciones

Acción	Definición informal	Definición formal
insertarSubclaseDe( $c_1, c_2, o_A$ )	Se inserta un enlace de <i>subclase de</i> entre los conceptos $c_1$ y $c_2$ de la ontología $o_A$ .	$subclaseDeO_{A_{i+1}} = subclaseDeO_{A_i} \cup \{(c_1, c_2)\}$
insertarConcepto( $c, o_A$ )	Se inserta el concepto $c$ en la ontología $o_A$ .	$C_{A_{i+1}} = C_{A_i} \cup \{c\}$
insertarAtributo( $a, c, o$ )	Se añade el atributo $a$ al concepto $c$ de la ontología $o$ .	$atributoDeO_{i+1} = atributoDeO_i \cup \{(a, c)\};$ $A_{A_{i+1}} = A_{A_i} \cup \{a\}$
insertarRelacion( $r, o_A$ )	Se inserta la relación $r$ en la ontología $o_A$ .	$R_{A_{i+1}} = R_{A_i} \cup \{r\}$
insertarRelacionHas( $c_1, c_2, o_A$ )	Se inserta la relación <i>tiene</i> entre los conceptos $c_1$ y $c_2$ en la ontología $o_A$ .	$R_{A_{i+1}} = R_{A_i} \cup \{tiene(c_1, c_2)\}$
borrarAtributo( $a, c, o_A$ )	Se borra el atributo $a$ del concepto $c$ de la ontología $o_A$ .	$atributoDeO_{A_{i+1}} = atributoDeO_{A_i} - \{(a, c)\};$ $A_{A_{i+1}} = A_{A_i} - \{a\}$
borrarRelacion( $r, o_A$ )	Se borra la relación $r$ en la ontología $o_A$ .	$R_{A_{i+1}} = R_{A_i} - \{r\}$
borrarSubclaseDe( $c_1, c_2, o_A$ )	Se borra el enlace de <i>subclase de</i> entre los conceptos $c_1$ y $c_2$ de la ontología $o_A$ .	$subclaseDeO_{A_{i+1}} = subclaseDeO_{A_i} - \{(c_1, c_2)\}$

El modelo OEGMerge, tanto de manera manual como automática (véase el apartado 5) se ha utilizado desde el año 2001 [3] [12] para mezclar ontologías en el Grupo de Ingeniería Ontológica. La implementación parcial del método, ODEMerge, está disponible para el uso de todos los usuarios de WebODE. Incluso los resultados de esta implementación se usan para trabajos externos, como [8].

### 5. Completud del modelo

Tras la presentación de los casos, queda por demostrar que cubren por completo las posibles situaciones, atendiendo a las dimensiones del tipo de componentes de las ontologías (conceptos, atributos y relaciones) y al tipo de mappings (similitud semántica, *subclase de* y *superclase de*).

Para ello, en la Tabla 5 se puede ver los casos que cubren el espectro de similitud entre los componentes.

Tabla 5: Casos de similitud entre los elementos

Similitud	Concepto base	Atributo base	Relación base
Concepto auxiliar	5	S-10	S-8
Atributo auxiliar	S-9	S-1	S-12
Relación auxiliar	S-7	S-11	S-4

Por ejemplo, como figura en la Tabla 5, la similitud entre concepto auxiliar y base está contemplada en el caso 5; la similitud entre la relación auxiliar y un atributo base está cubierta por el caso S-11.

Los casos en que los elementos no tienen similitud con ningún otro aparecen en la Tabla 6.

Tabla 6: Casos sin similitud

	Sin similitud
Concepto base	4
Concepto auxiliar	3
Atributo base	S-2
Atributo auxiliar	S-3
Relación base	S-5
Relación auxiliar	S-6

Así, por ejemplo, si un concepto base no es similar a ningún elemento auxiliar está reconocido en el caso 4, y el que un atributo auxiliar no sea similar a ningún elemento base está cubierto por el caso S-3. Las situaciones de diferencias taxonómicas están divididas en tres tablas (Tabla 7, Tabla 8 y Tabla 9) con tres pares de comparaciones: *subclase de* entre conceptos, *subclase de* transitiva entre antecedentes de conceptos similares y *subclase de* entre padres y hermanos de conceptos similares. Las combinaciones de similitud entre conceptos de las dos ontologías ya están consideradas en el caso 5 y las de sin similitud están consideradas en los casos 3 y 4.

Tabla 7: Casos de mapping *subclase de* simples

	Mapping = <i>subclase de</i>
Concepto base relacionado con concepto auxiliar	1
Concepto auxiliar relacionado con concepto base	2

**Tabla 8: Casos de similitud entre padres y hermanos**

<i>Similitud</i>	Padre de concepto auxiliar	Hermano de concepto auxiliar
Padre de concepto base	<b>S-13</b>	<b>S-16</b>
Hermano de concepto base	<b>S-15</b>	<b>Ya considerado en 5</b>

**Tabla 9: Casos sin similitud entre padres y hermanos**

<i>Sin Similitud</i>	Padre de concepto auxiliar	Hermano de concepto auxiliar
Padre de concepto base	<b>S-14</b>	<b>Ya considerado en 3 y 4</b>
Hermano de concepto base	<b>Ya considerado en 3 y 4</b>	<b>Ya considerado en 3 y 4</b>

## 6. ODEMerge

Con la realización de numerosas mezclas de ontologías siguiendo el modelo OEGMerge y utilizando la técnica descrita en el apartado 4, apareció la necesidad de automatizar el proceso de detección de condiciones y aplicación de acciones del método que resuelve de mezcla, ahorrando al ingeniero ontológico un esfuerzo considerable. Para ello, se desarrolló un servicio integrado en la plataforma WebODE: ODEMerge.

ODEMerge [12] es un servicio que realiza la mezcla automática de las taxonomías de dos ontologías a partir de las ontologías y de dos tablas, que generará el ingeniero ontológico, con información semántica sobre sinonimia e hiperonimia entre los conceptos de las ontologías (información semántica que tiene como entrada el servicio equivalen a la información aportada por el experto del dominio en el método, es decir, el conjunto de *mappings*). Los detalles referentes a formato de las tablas y limitaciones del servicio se pueden encontrar en la ayuda de la plataforma y en [12].

Así, ODEMerge pretende realizar, en la medida de lo posible y de una manera automática, las acciones de la mezcla taxonómica del modelo OEGMerge. La automatización conlleva la eliminación de la copia de una ontología para tomarla como base, puesto

que ya no se trabaja sobre una de las ontologías para producir la unificada, sino que la ontología unificada se genera independientemente. Además, en la última versión de este servicio se ha incorporado a la automatización la inserción de atributos de clase sobre el componente de la ontología original del que proviene el concepto, lo que proporciona trazabilidad al proceso mezcla.

## 7. Conclusiones y trabajos futuros

Con el trabajo aquí presentado mediante el modelo OEGMerge, se establece de forma explícita la casuística en la mezcla de ontologías que constan de conceptos, atributos y relaciones, y se establece las acciones resolutorias para llevar a cabo el proceso para el caso reducido de conceptos, atributos y relaciones (tres de los componentes fundamentales de toda ontología bajo el paradigma de marcos). Además, se ha citado el software desarrollado para la automatización de parte del modelo sobre una plataforma de ingeniería ontológica como es WebODE, cubriendo así otra tarea más del ciclo de vida de las ontologías. Tanto el modelo como el software presentado se han probado desde su desarrollo en numerosos procesos de mezcla llevados a cabo por el Grupo de Ingeniería Ontológica de la UPM.

Queda para un trabajo futuro el seguir ampliando tanto OEGMerge como ODEMerge para conseguir abarcar el resto de componentes de ontologías que ahora no abarca (axiomas, referencias, grupos, etc.) tanto bajo el paradigma de marcos como otros. Además, como se ha indicado ya, habría que contemplar en la definición de los *mappings* predicados nuevos para expresar correspondencias con más semántica.

## Agradecimientos

La formalización del modelo ha sido financiada por el proyecto Infraestructura tecnológica de servicios semánticos para la web semántica (TIN-2004-02660), del Ministerio de Ciencia y Tecnología. También agradecemos los comentarios realizados a María del Carmen Suárez de Figueroa Baonza y Miguel Esteban Gutiérrez.



## Referencias

- [1] Arpírez JC, Corcho O, Fernández-López M, Gómez-Pérez A. WebODE in a nutshell. *AI Magazine*, 24(3)-37-47
- [2] Cañadas G, Fernández-López M, García-García R, Lama M, Sánchez-Alberca A, Sorzano COS. Framework for automatic generation of ontology mappings. In: *Dieste O (ed) Jornadas Iberoamericanas de Ingeniería del Software e Ingeniería del Conocimiento (JIISIC'04)*. Madrid, Spain
- [3] De Diego R. Método de mezcla de catálogos electrónicos. *Proyecto final de carrera, Facultad de Informática, Universidad Politécnica de Madrid*. 2001.
- [4] Doan A, Madhavan J, Domingos P, Halvey A. Learning to map between ontologies on the semantic web. In *Proceedings of 11th International WWW Conference*, Hawaii (2002)
- [5] Ehrig M, Sure Y. Ontology mapping - an integrated approach. In *Christoph Bussler, John Davis, Dieter Fensel, and Rudi Studer, editors, Proceedings of the First European Semantic Web Symposium, volume 3053 of Lecture Notes in Computer Science*, pages 76--91, Heraklion, Greece, 2004. Springer Verlag.
- [6] Gangemi A, Pisanelli DM, Steve G. An Overview of the ONIONS Project: Applying Ontologies to the Integration of Medical Terminologies. *Data & Knowledge Engineering* 31(2):183-220
- [7] Guarino N, Welty C. A Formal Ontology of Properties. In: *Dieng R, Corby O (eds) 12th International Conference in Knowledge Engineering and Knowledge Management (EKAW'00)*. Juan-Les-Pins, France. (Lecture Notes in Artificial Intelligence LNAI 1937) Springer-Verlag, Berlin, Germany, pp 97-112
- [8] Gutiérrez L. Ontology-based semantic querying of the Web with respect to food recipes. *IMM-Thesis. Technical University of Denmark*. ISSN: 1601-233X
- [9] Madhavan J, Bern-stein PA, Rahm E. Generic schema matching with Cupid. *VLDB Journal* (2001) 49-58
- [10] Noy NF, Musen MA. SMART: Automated Support for Ontology Merging and Alignment In: *Gaines BR, Kremer B, Musen MA (eds) 12th Banff Workshop on Knowledge Acquisition, Modeling, and Management, Banff, Alberta, Canada*, 4-7:1-20
- [11] Noy NF, Musen MA. PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment *Proceedings of the Seventeenth National Conference on Artificial Intelligence (AAAI-2000)*. Austin, Texas. 2000.
- [12] Ramos JA. Mezcla automática de catálogos electrónicos. *Proyecto final de carrera, Facultad de Informática, Universidad Politécnica de Madrid*. 2001.
- [13] Studer R, Benjamins VR, Fensel D. Knowledge Engineering: Principles and Methods. *IEEE Transactions on Data and Knowledge Engineering* 25(1-2):161-197.
- [14] Stumme G, Maedche A. FCA-Merge: Bottom-Up Merging of Ontologies *IJCAI 2001 - Proceedings of the 17th International Joint Conference on Artificial Intelligent*. Seattle, USA, Agosto 2001.